

Code City →

Richard Wettel and Michele Lanza

University of Lugano, CH



Software systems as

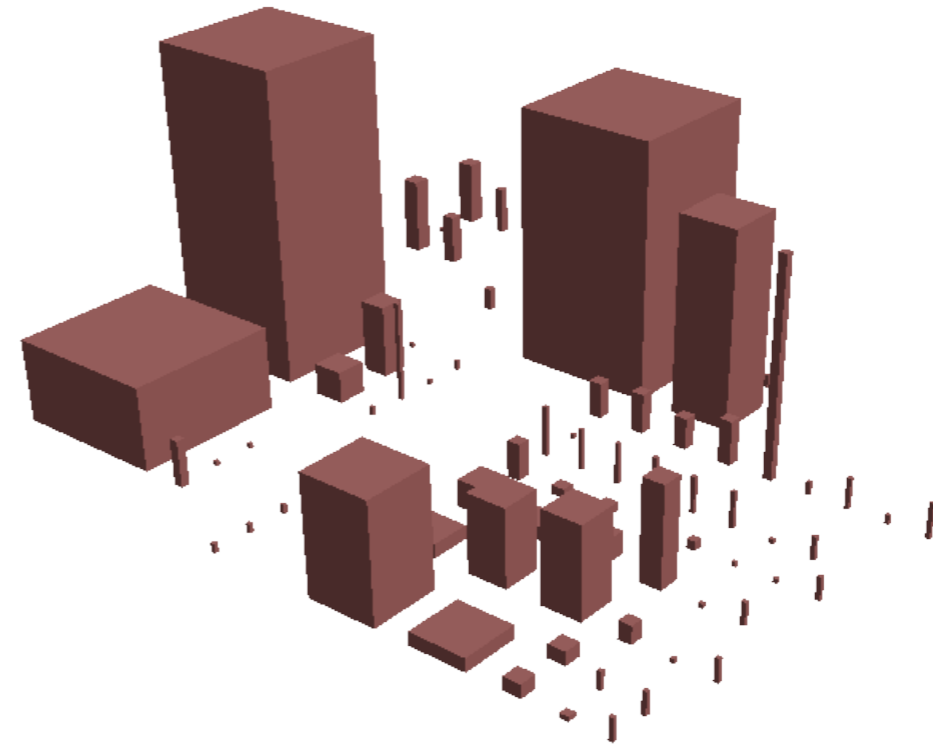
The city metaphor

The city metaphor

domain mapping	

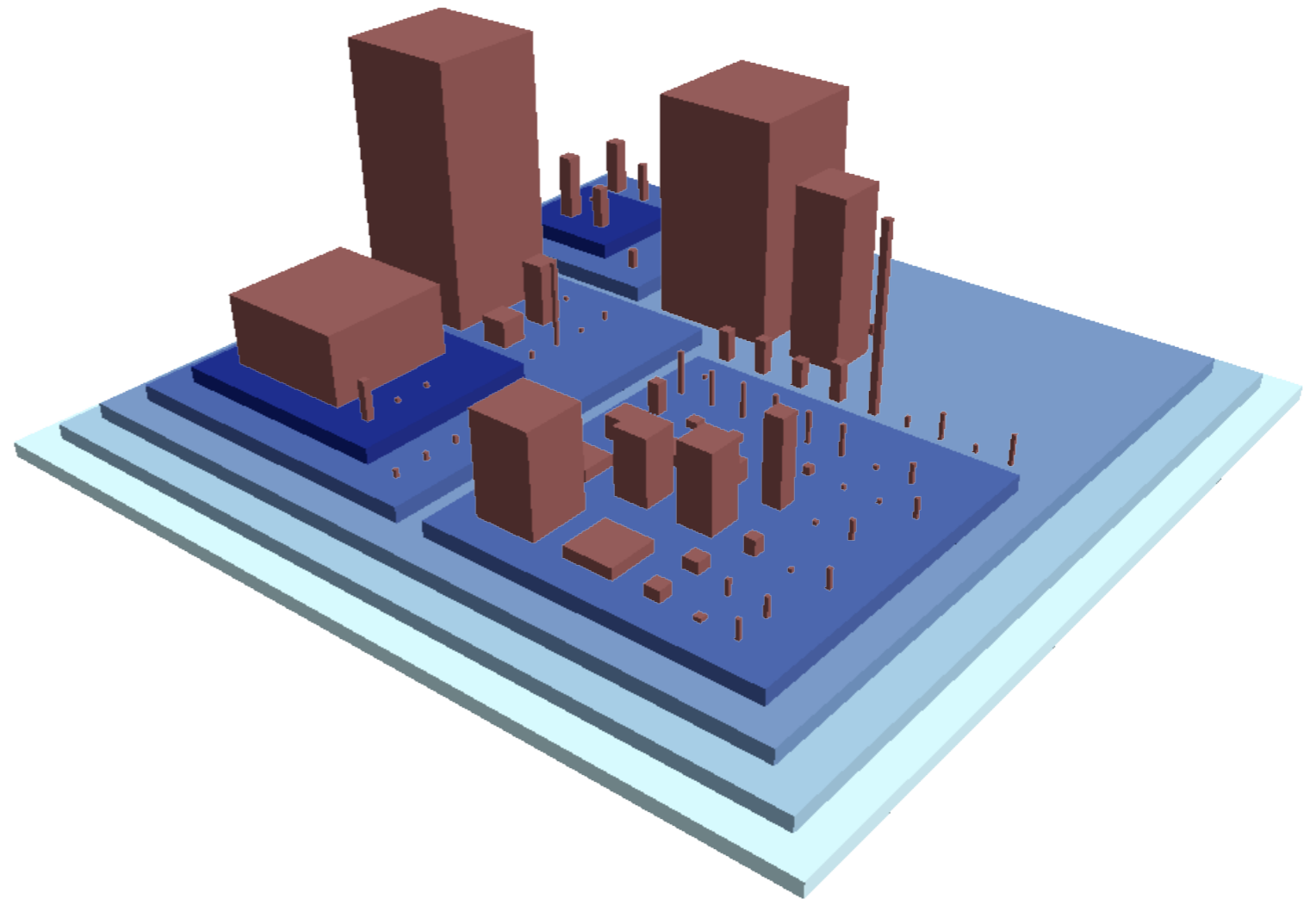
The city metaphor

domain mapping	
classes	buildings



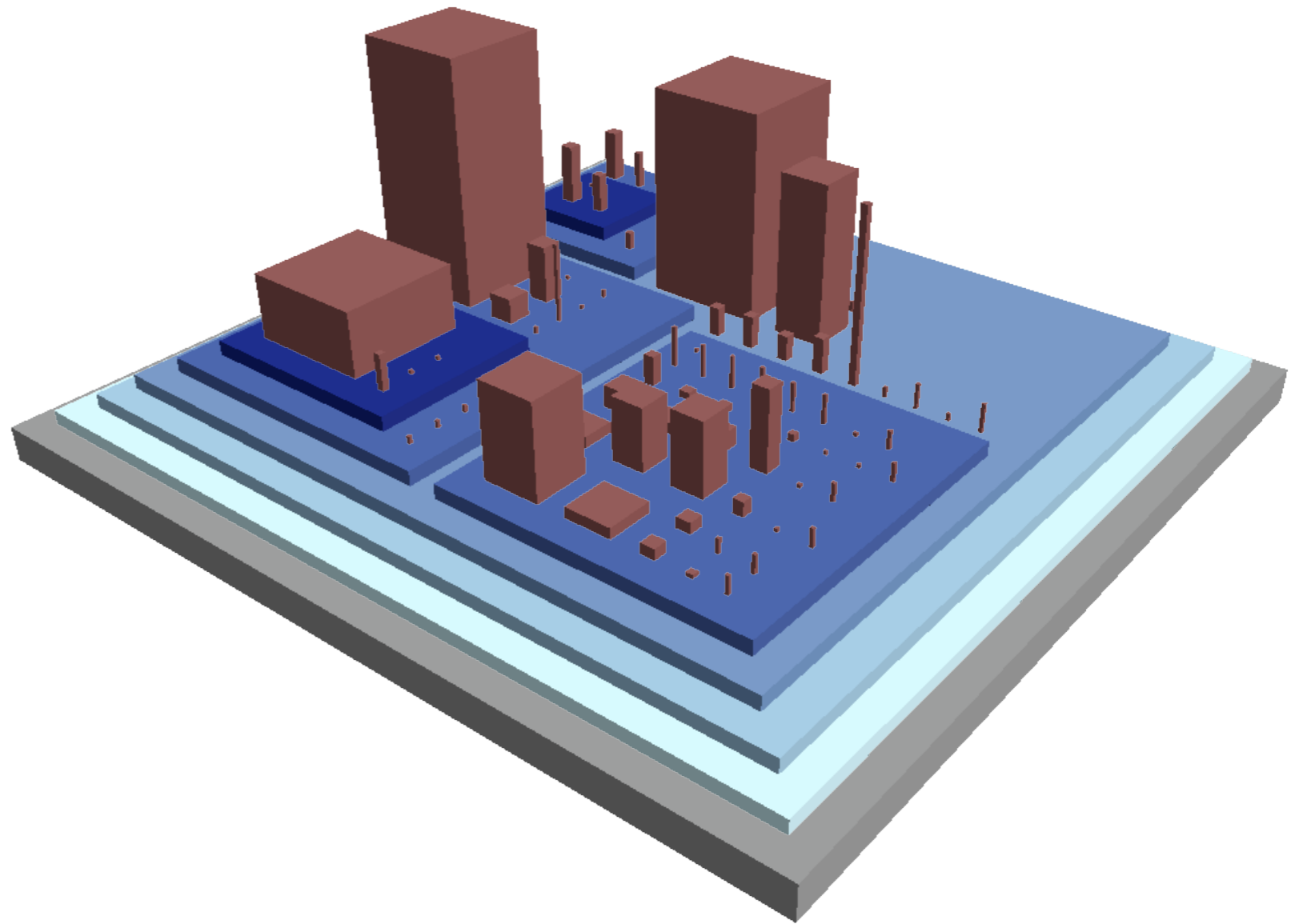
The city metaphor

domain mapping	
classes	buildings
package	districts



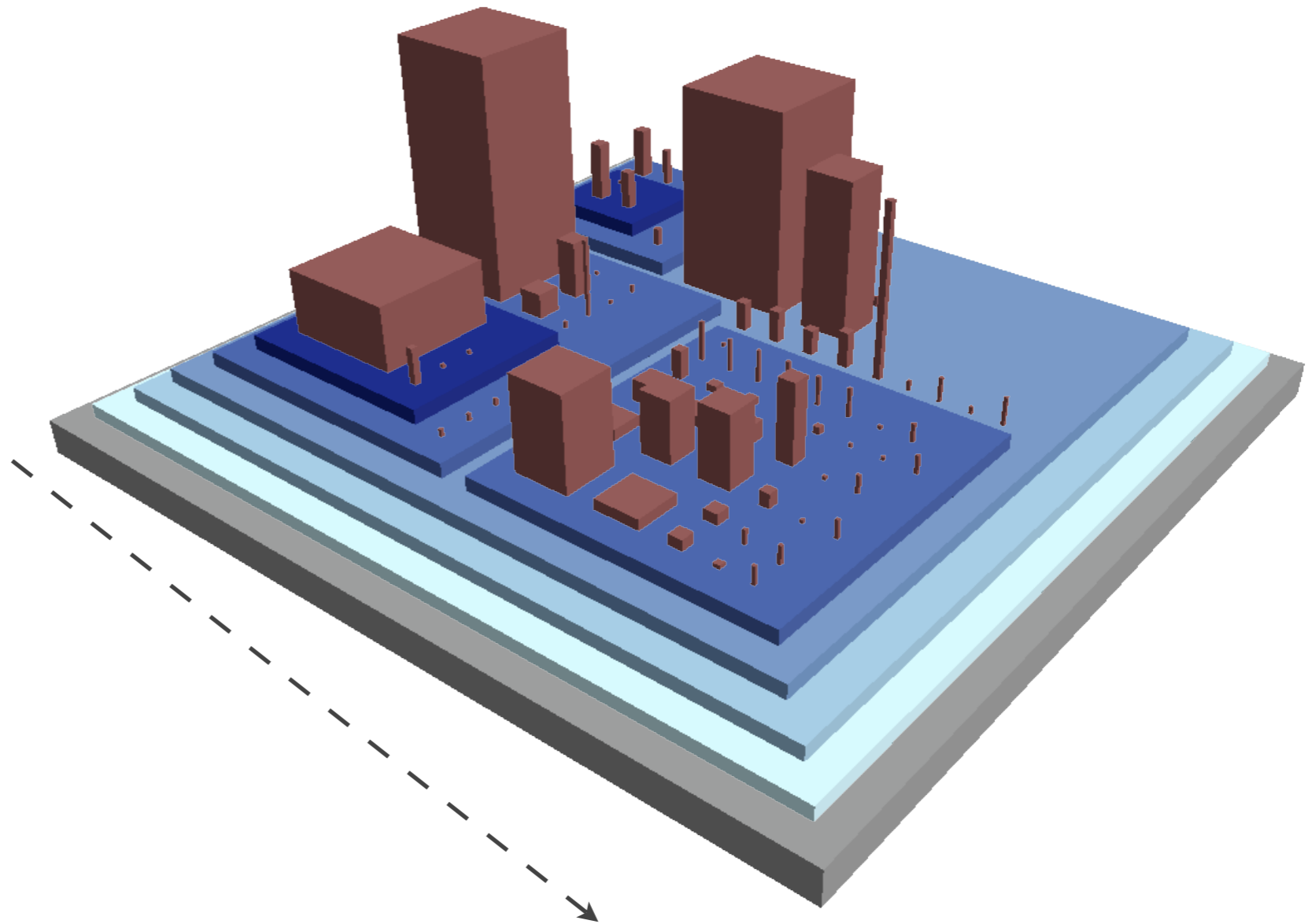
The city metaphor

domain mapping	
classes	buildings
package	districts
system	city



The city metaphor

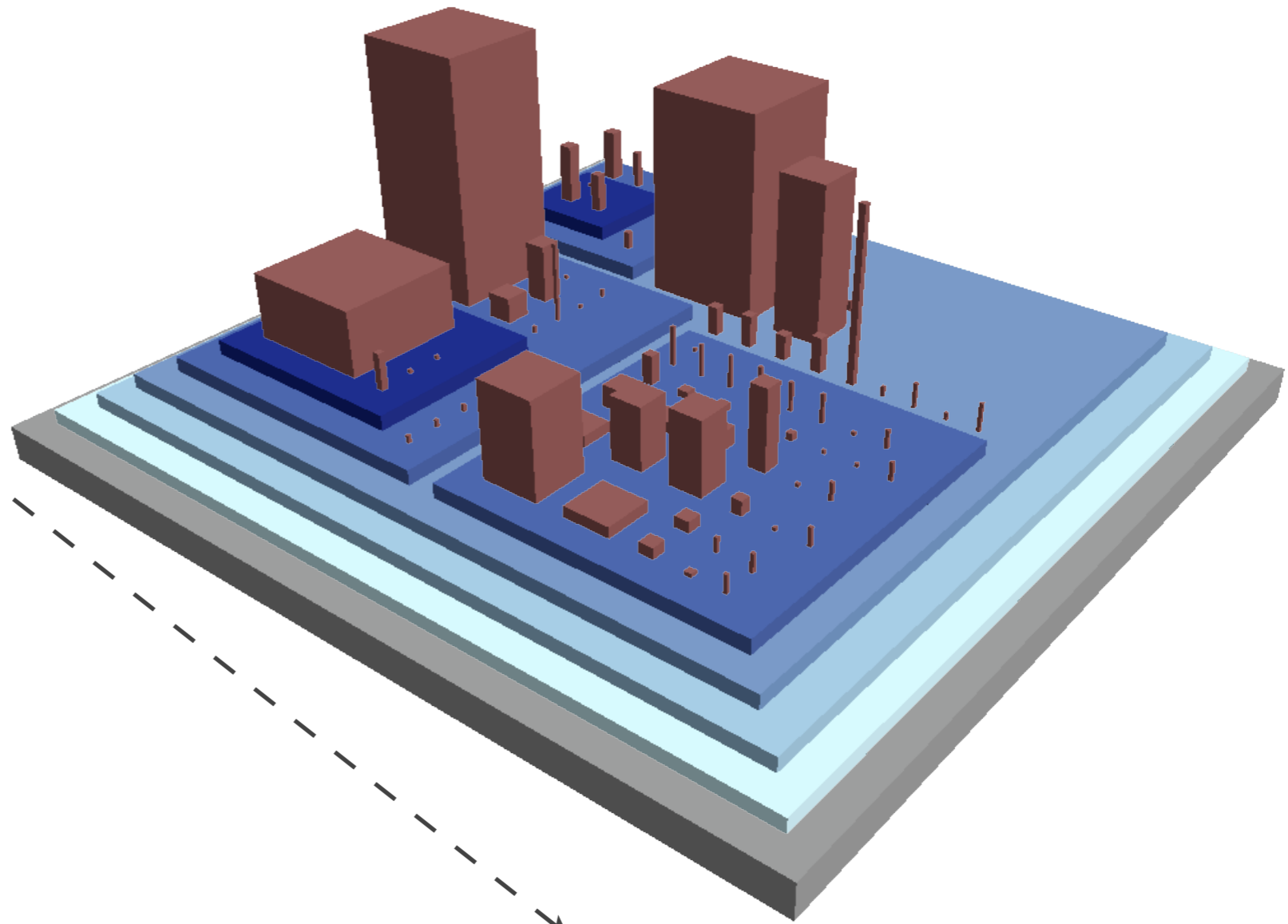
domain mapping	
classes	buildings
package	districts
system	city



class metric	building
number of	height
number of	width,

The city metaphor

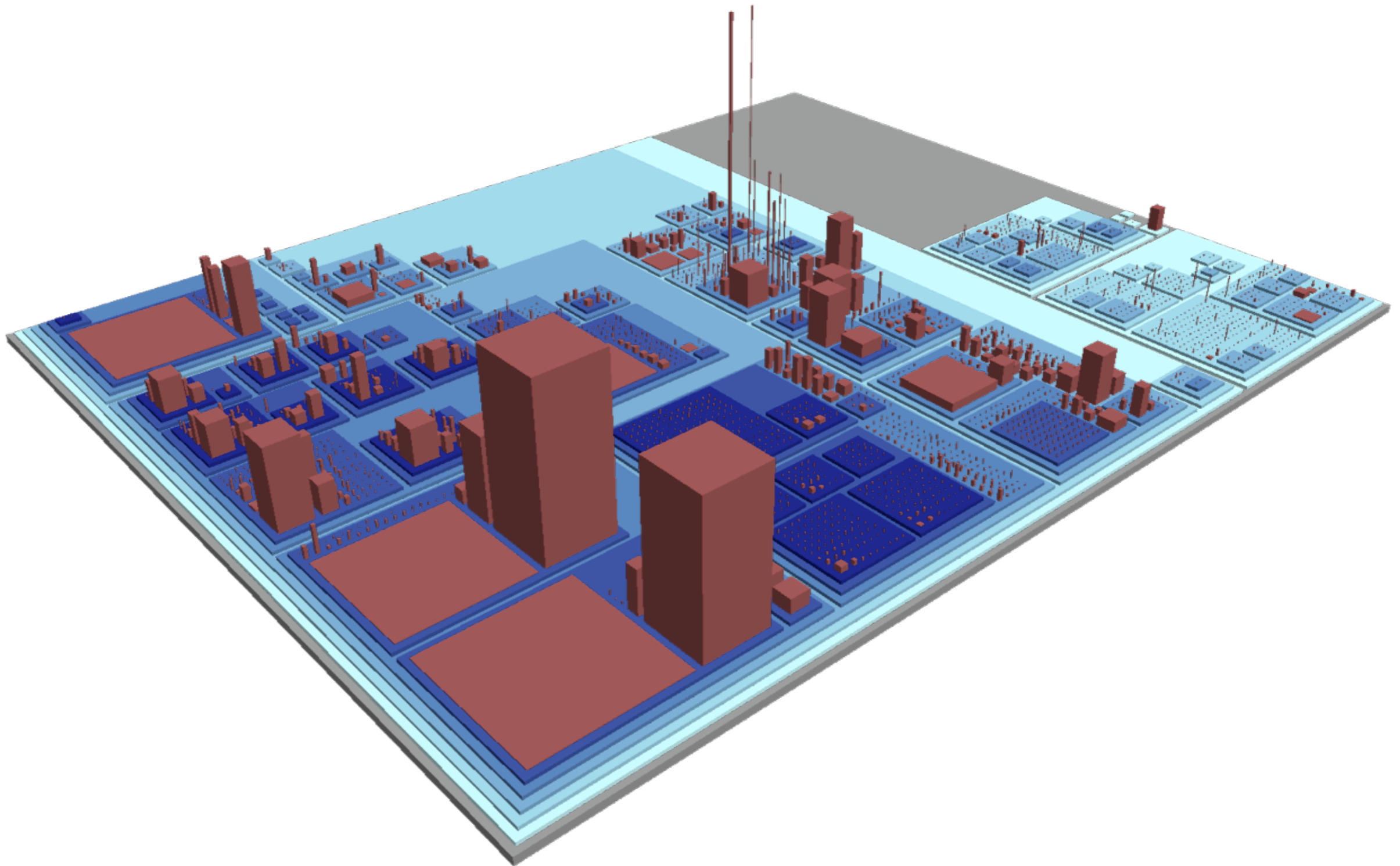
domain mapping	
classes	buildings
package	districts
system	city



package	district
nesting level	color

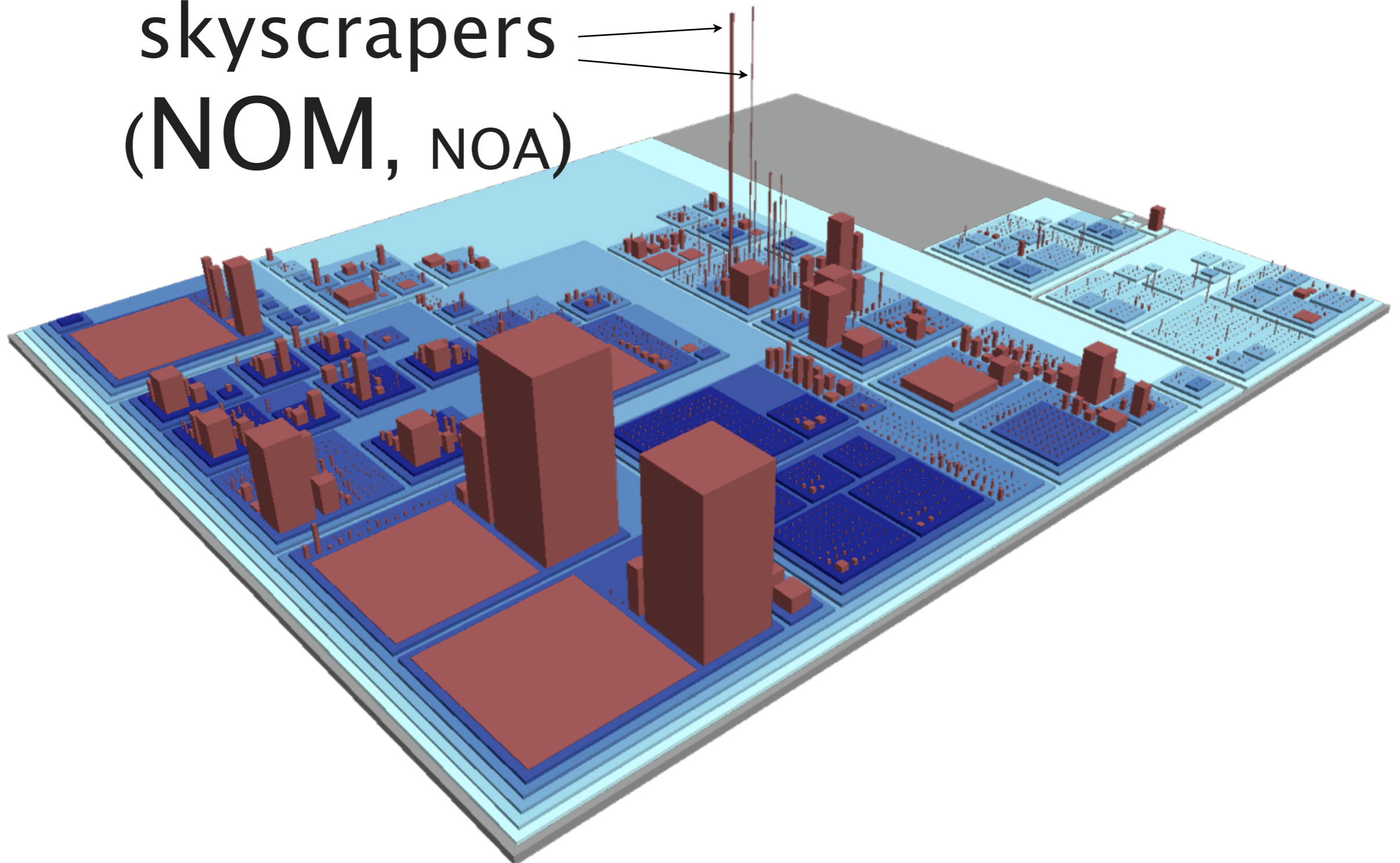
class metric	building
number of	height
number of	width,

“Reading” a code city



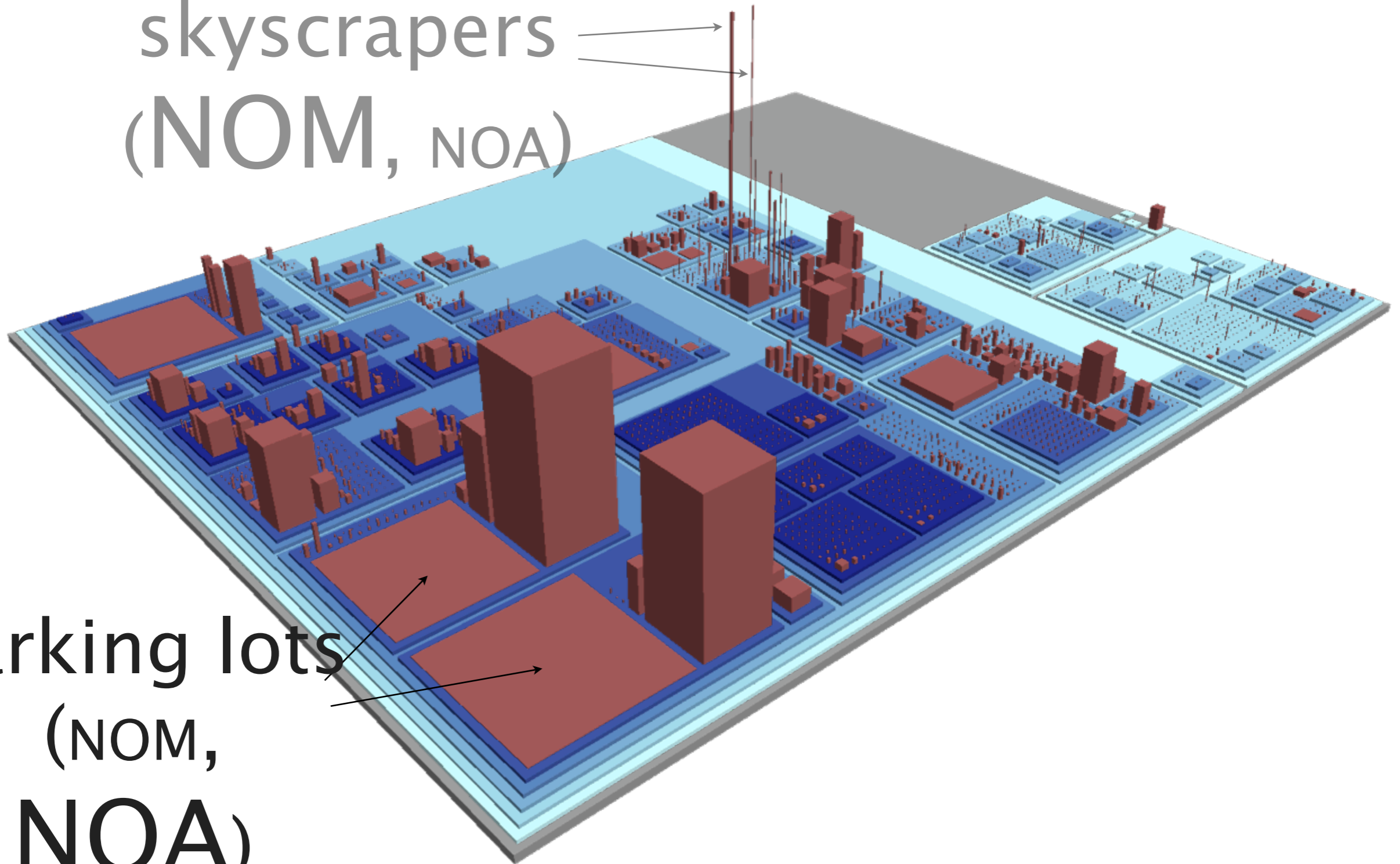
“Reading” a code city

skyscrapers
(NOM, NOA)



“Reading” a code city

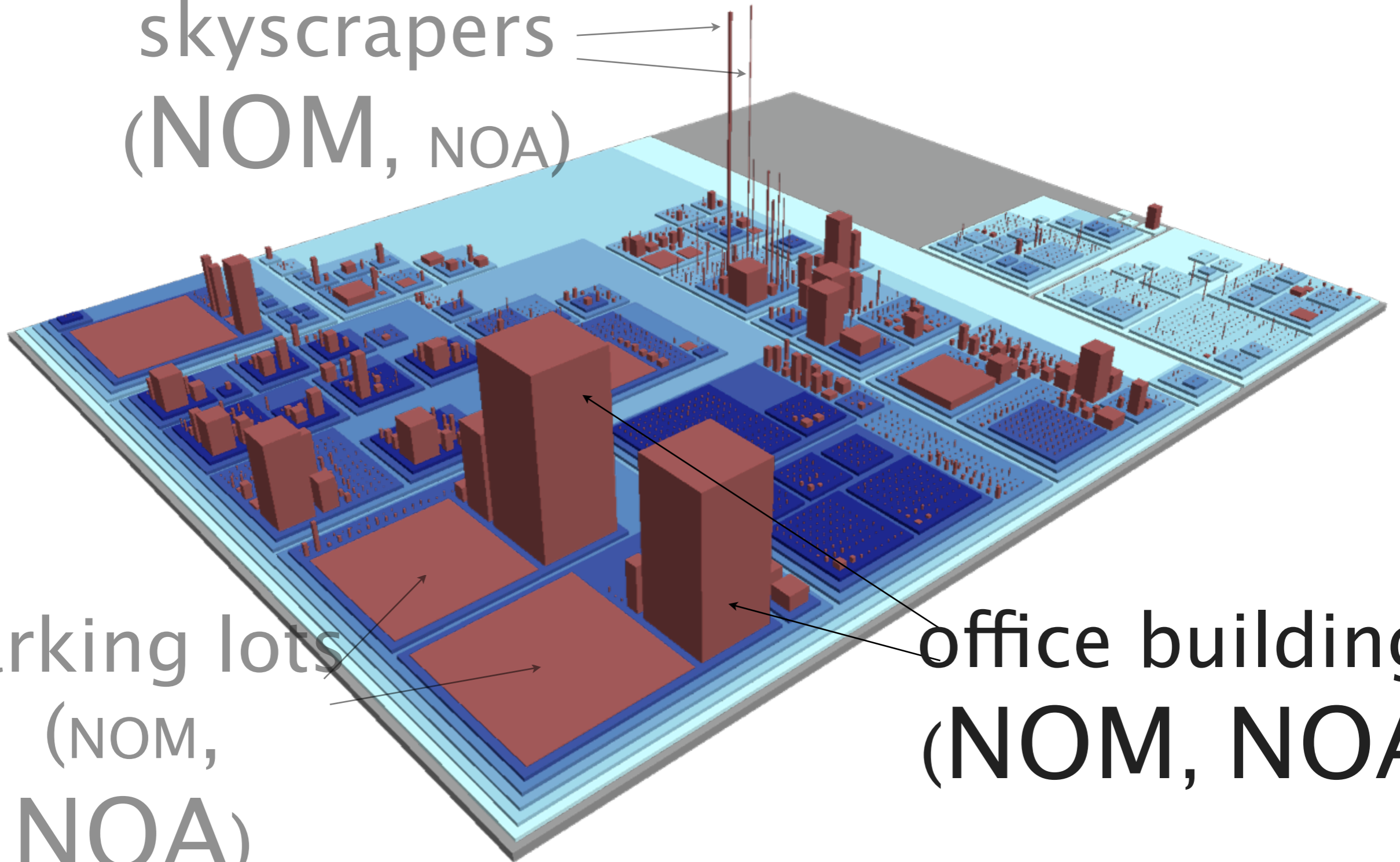
skyscrapers
(NOM, NOA)



parking lots
(NOM,
NOA)

“Reading” a code city

skyscrapers
(NOM, NOA)



parking lots
(NOM,
NOA)

office buildings
(NOM, NOA)

TABULA GEOGRAPHICA
 utriusque
 HEMISPHERII TERRESTRIS

exhibens
 declinationem acus magneticæ
 pro singulis locis globi terræque ad A.C. 1744.
 Jussu
 A.C.A.D. REG. SCIENT. et ED. LITT. BOR.
 descripta.



Com
 Declinatio
 acus magneticæ
 in iisdem terra locis
 successu temporis sit mutabilis
 Hæcque prius ex observationibus
 undique collectis hujusmodi mappam ad A. 1700

Singula linea que in hac tabula conspiciuntur ita sunt ductæ ut per omnia terra loca in quibus A. 1744 acus magnetica eandem declinationem est passu transierat ita primum linea tractu duplicato ducta ubi declinatio erat nulla designat. Ea ex observationibus australibus per mare atlanticum ascendens Americam supra tropicum canem transiit ac mari pacifico percurso iuxta Japoniam lit-
 tora progressa mare fluxu per Indiam orientalem revertitur in regionem australes unde sine dubio via adhuc ignota in mari atlantico emergit. Si que lineam continuam in se redeuntem conspiciunt videtur extra hanc lineam acus magnetica ubique declinat sive orientem sive occidentem versus prout tabula in spectu declarat ubi declinationes orientales littera P. Occiden-
 tales vero littera V. sunt indicata.

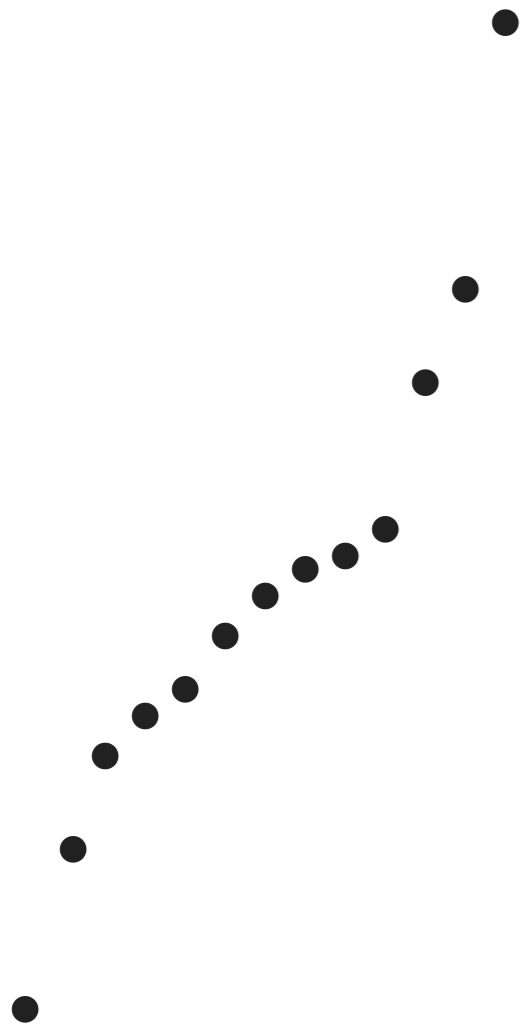
secundum legem chartarum nauticarum continent. Deinde vero similis mappa a Gulielmo Mountaine et Jacobo Dodson in Anglia est edita, qua variationes magneticas ad A. 1744 representat. Quoniam autem in usitata projectione verus tractus linearum que per lo-
 ca eandem declinationis dantur diffusius perspicitur, vi-
 sum est eandem hanc mappam more conato super ambobus terra hemispherii exhibere, quo uno quasi obtutu nexus
 et ordo harum linearum, facilius agnosci queat.

In hemispherio superiori tractus harum linearum præcipue au-
 tum versus satie regularis videtur cursu salicet recta secundum
 meridiana ad polum antarcticum directo. Interim tamen ea na-
 tura rei fieri nequit ut illa harum linearum in polum usque por-
 rigatur cum enim sub ipsis polis distinctis plagarum mundi punctis
 sit quomocunque ibi directionem acus magnetica tenuerit, nulla certa
 declinatio ei tribui poterit hæcque ob causam continuatis harum
 linearum polos versus plane est ignota. In hemispherio etiam in-
 terius deprehenditur neque ullo modo patet, quomodo in parte occiden-
 tali linea incerta sine continuanda se in manifesta incommoda de-
 libamus. Circa polum borealem autem omnia adhuc multo
 magis sunt incerta solo Hudsonis sine excepto.

Mapping techniques

1. Identity mapping

NOM



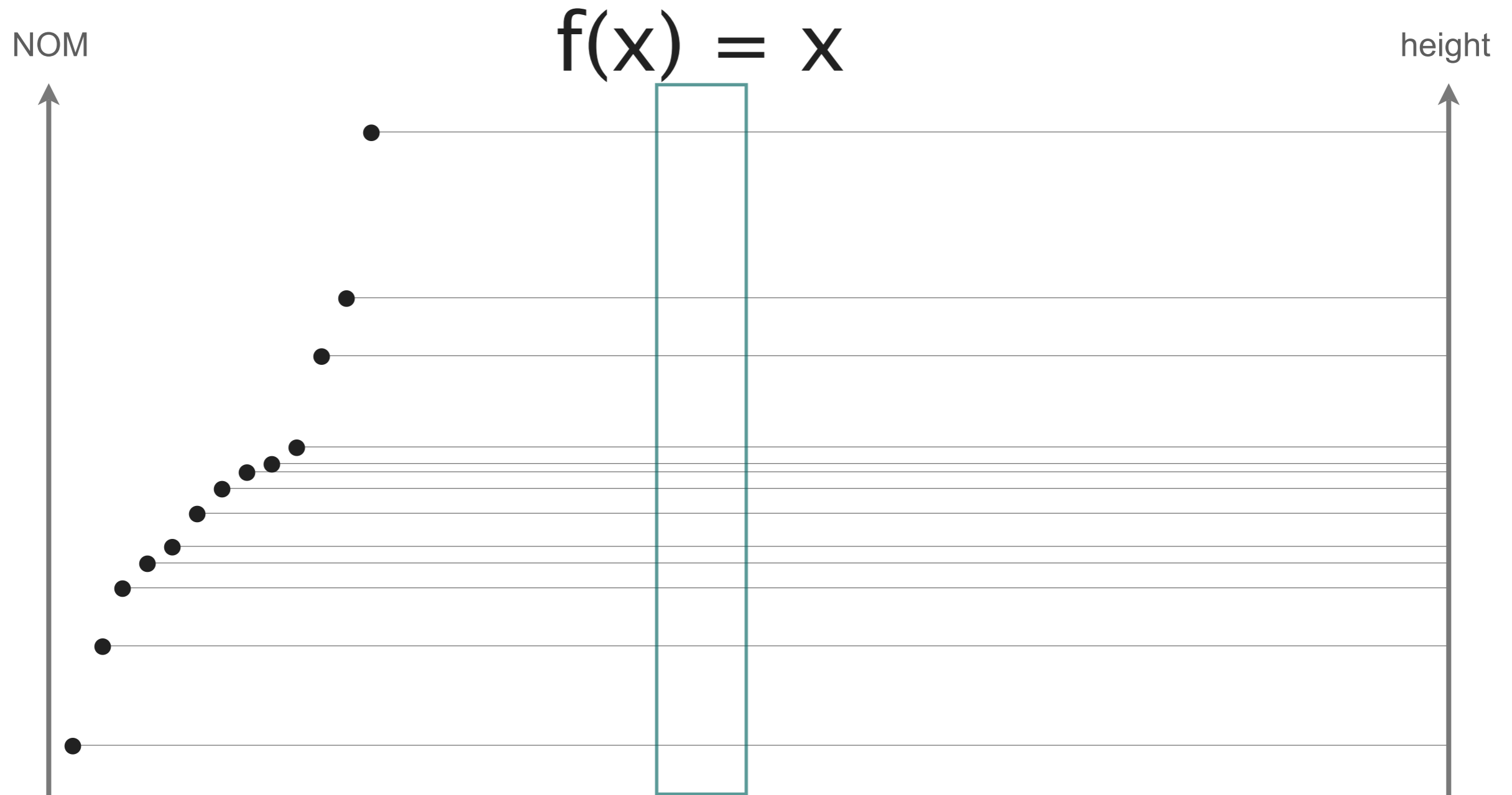
$$f(x) = x$$



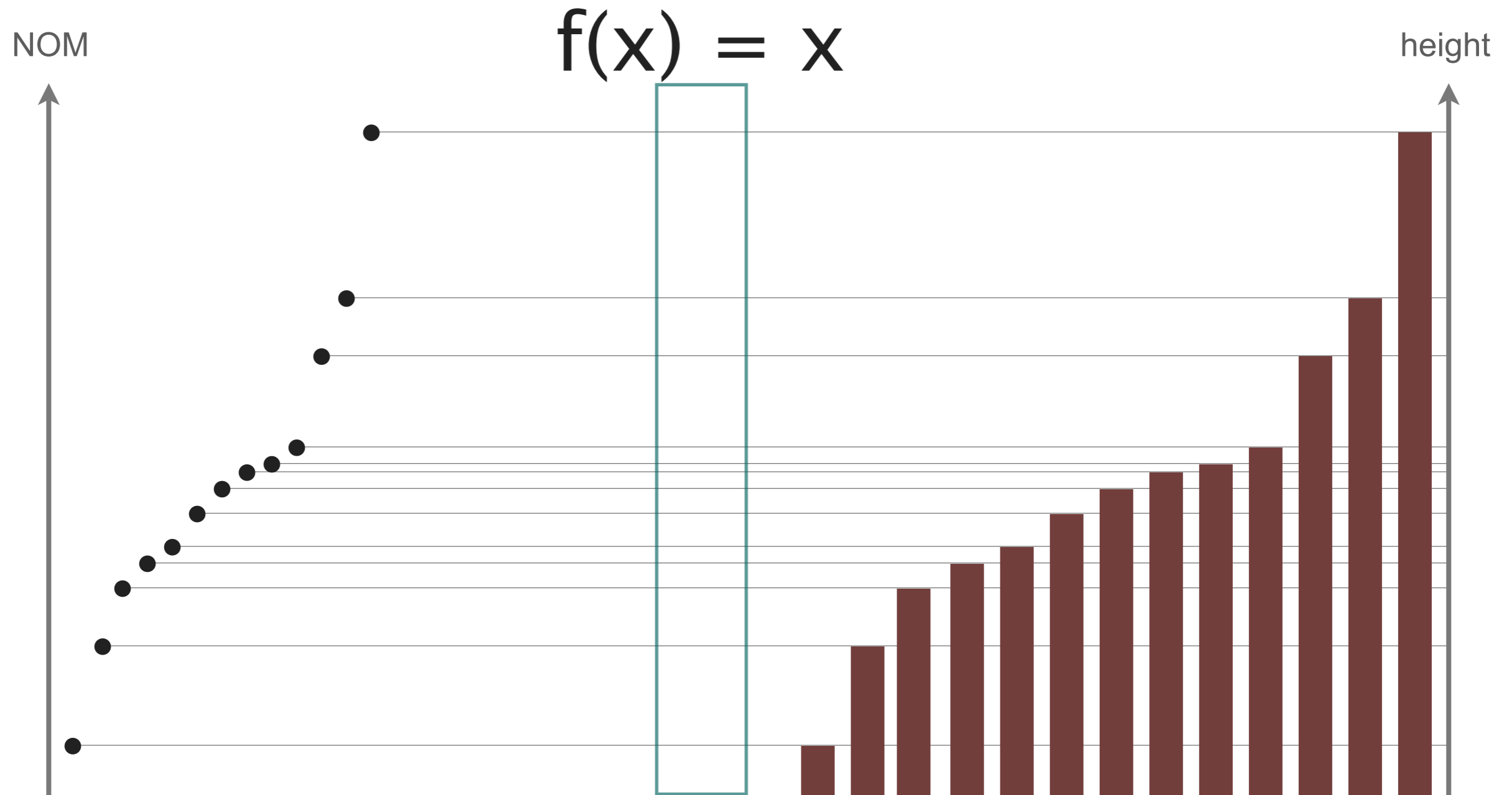
height



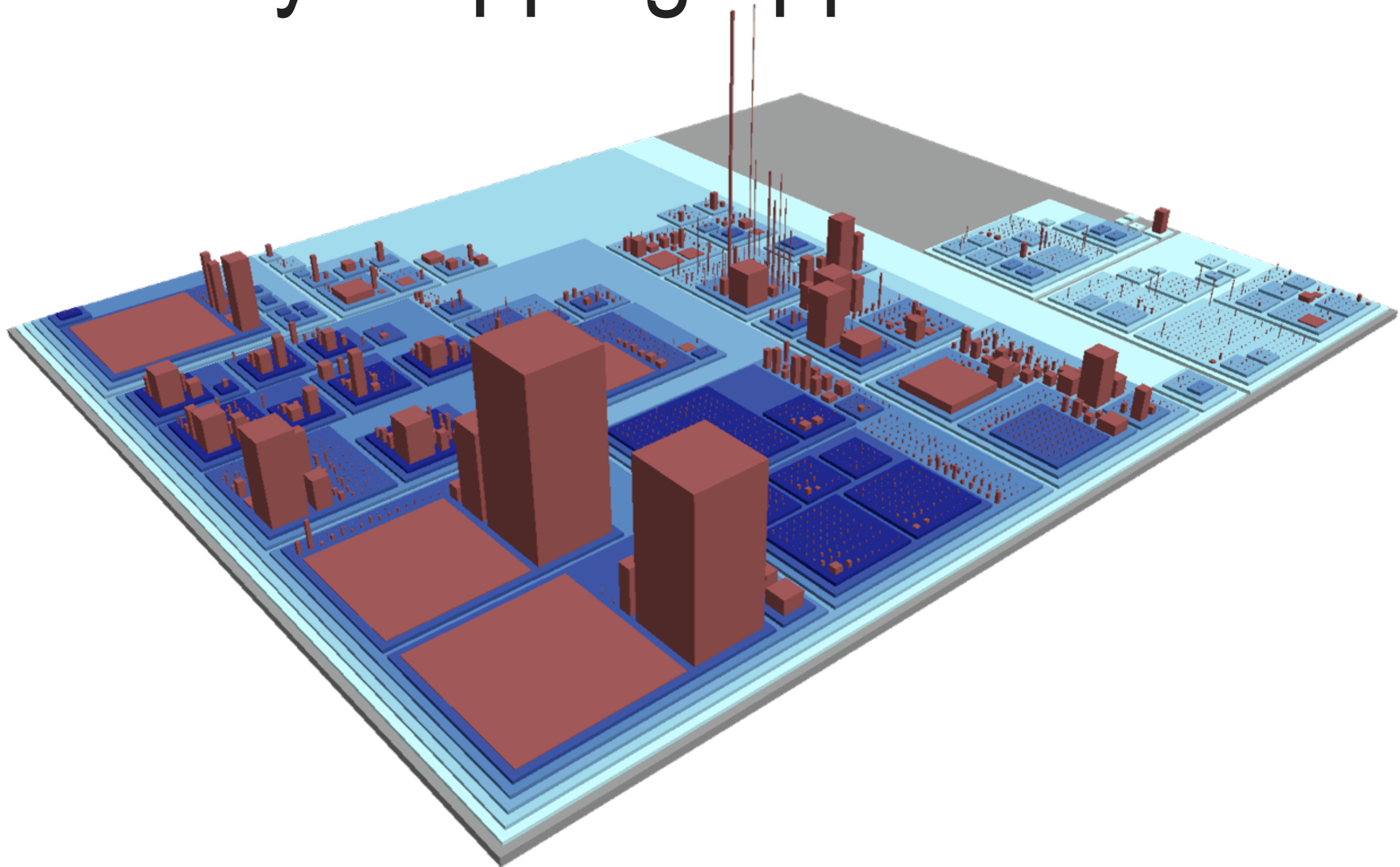
1. Identity mapping



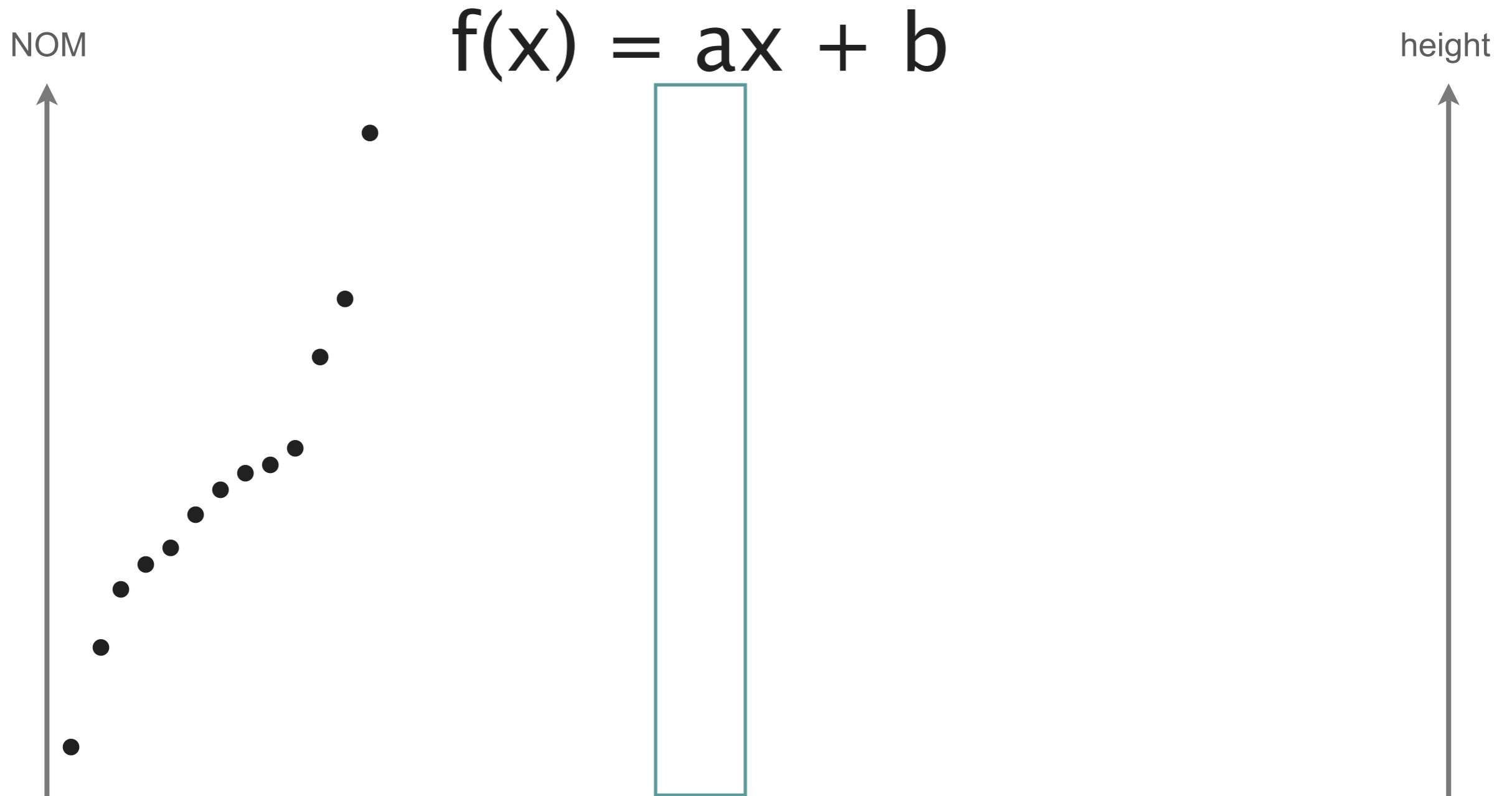
1. Identity mapping



Identity mapping applied

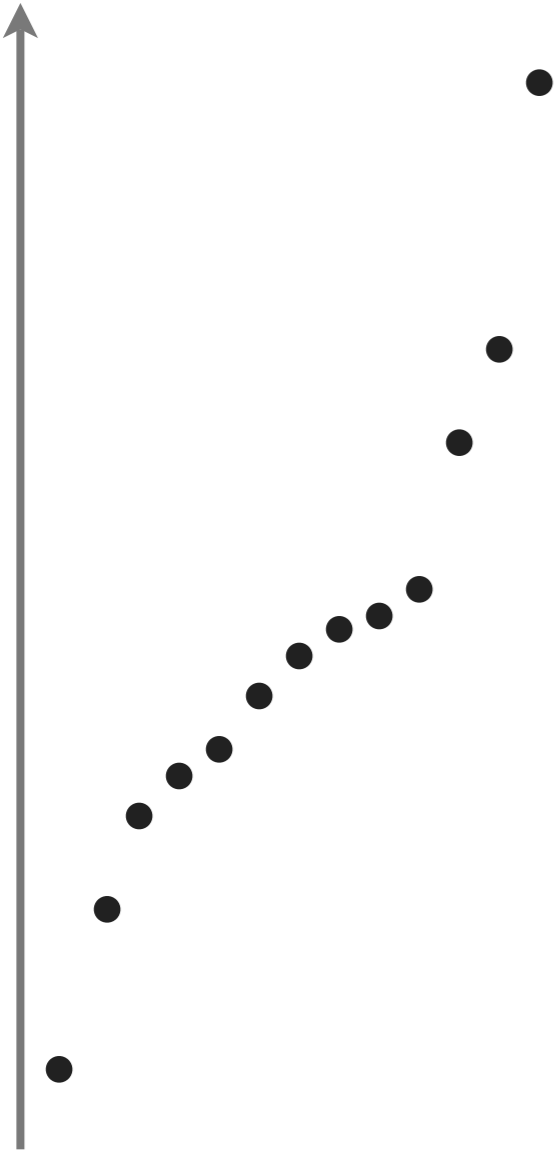


2. Linear mapping



2. Linear mapping

NOM



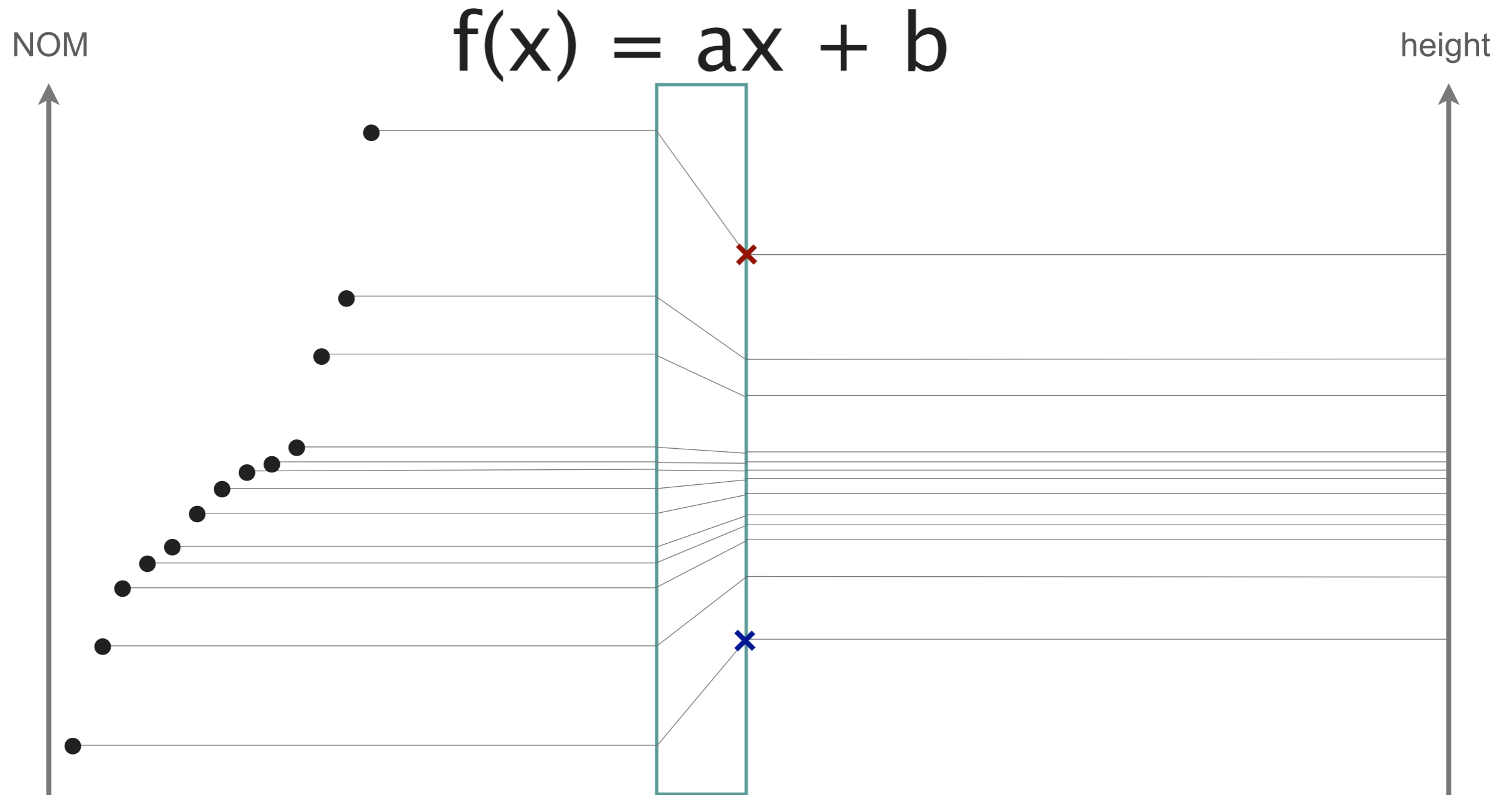
$$f(x) = ax + b$$



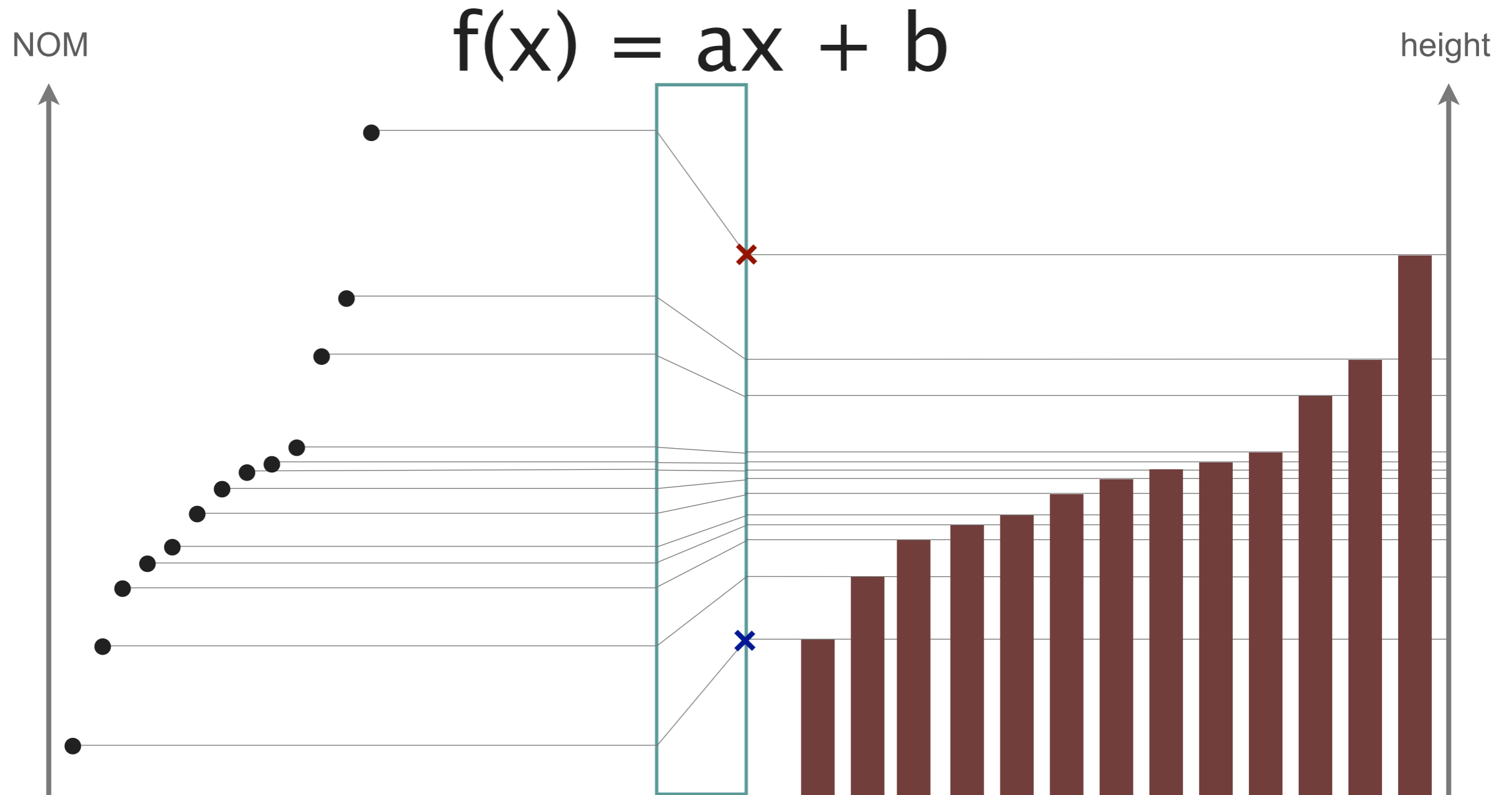
height



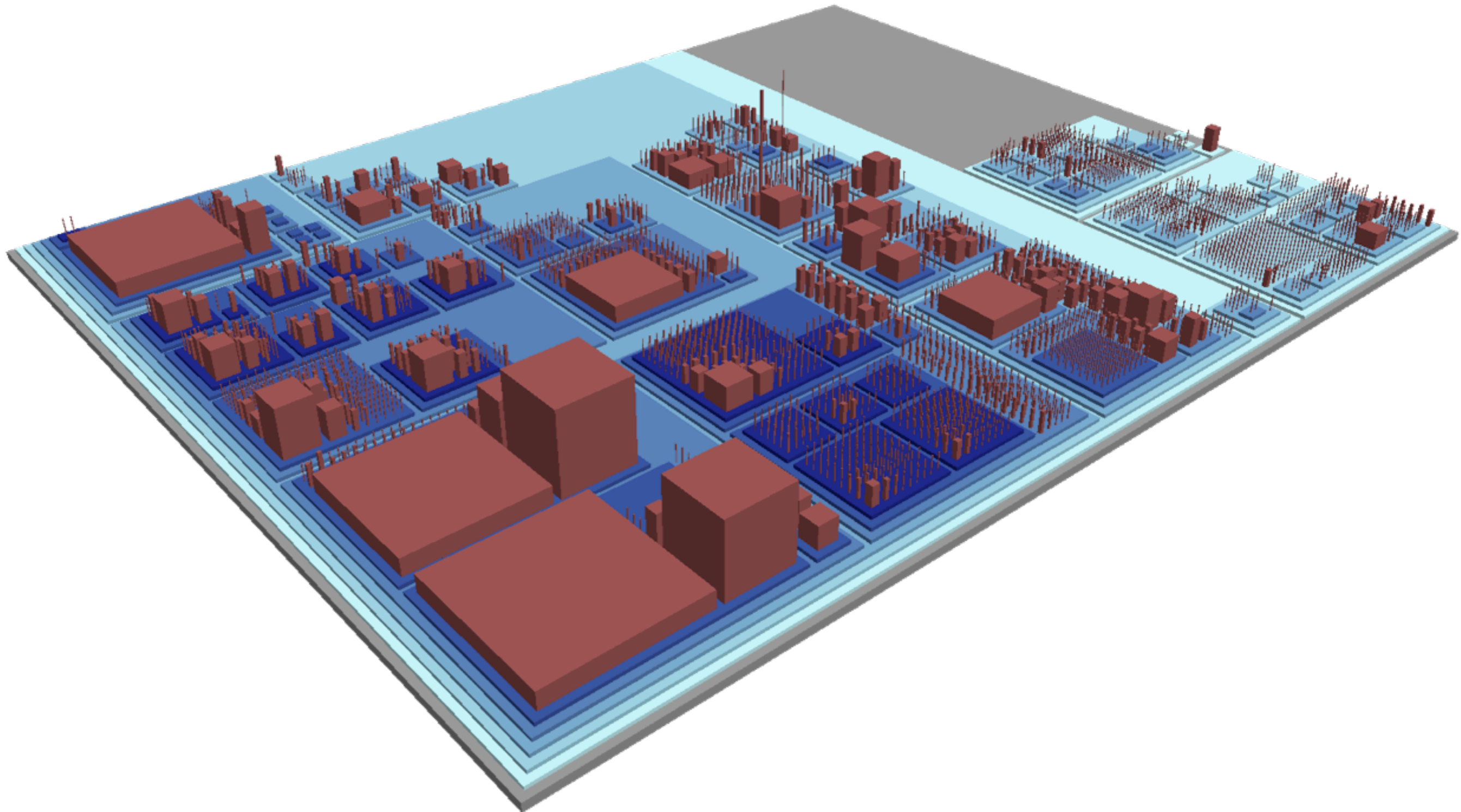
2. Linear mapping



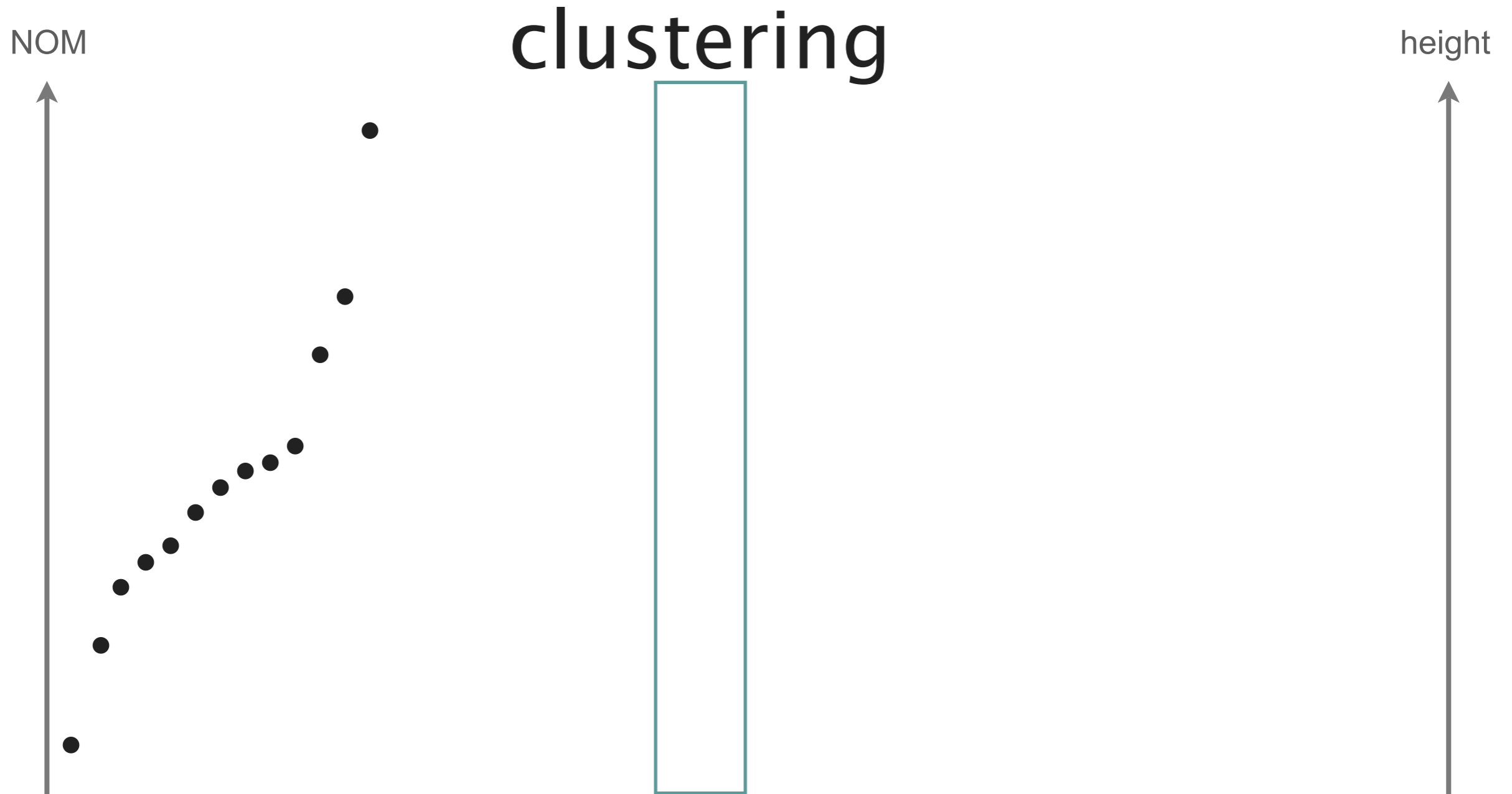
2. Linear mapping



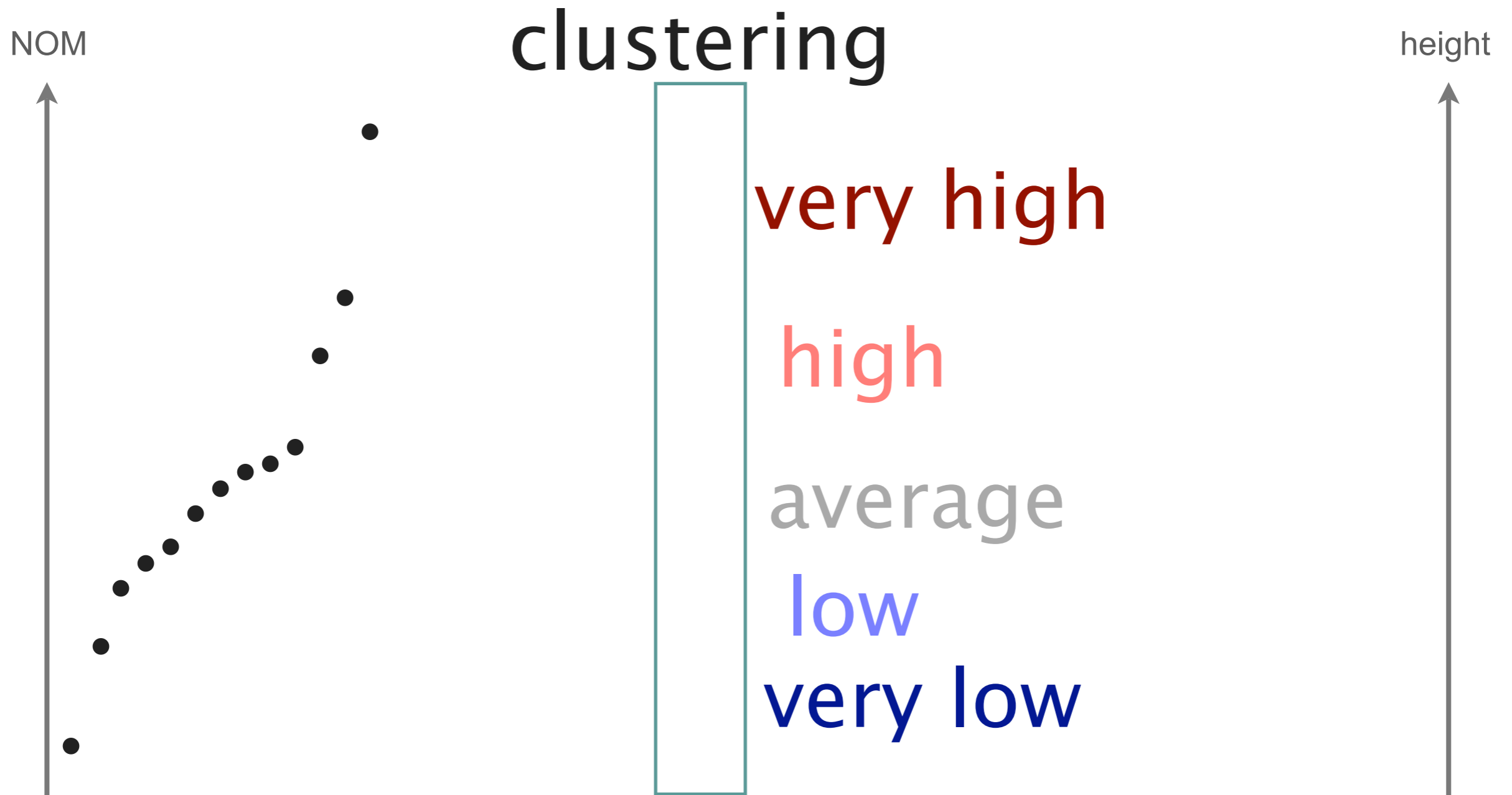
Linear mapping applied



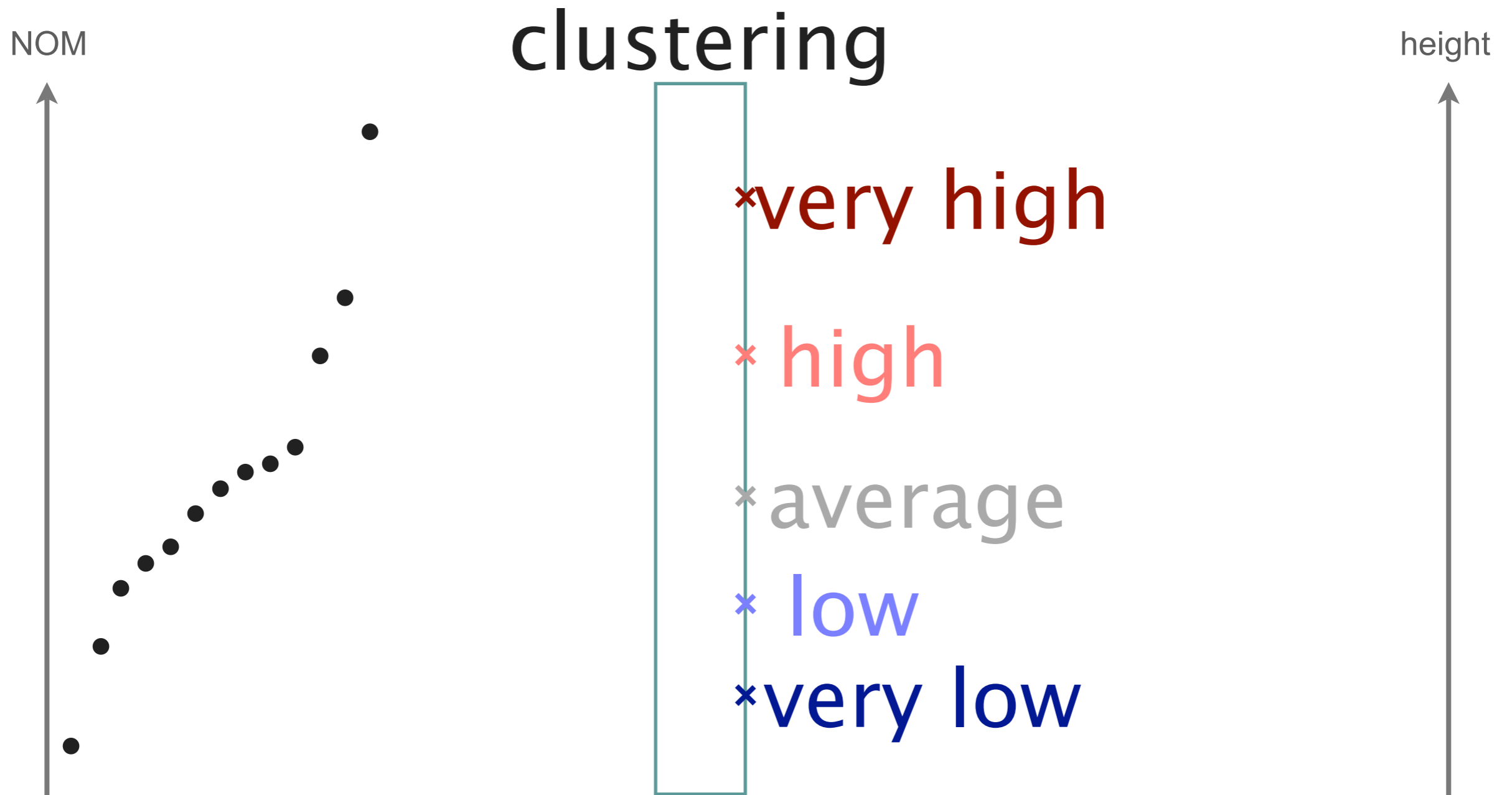
3. Boxplot-based mapping



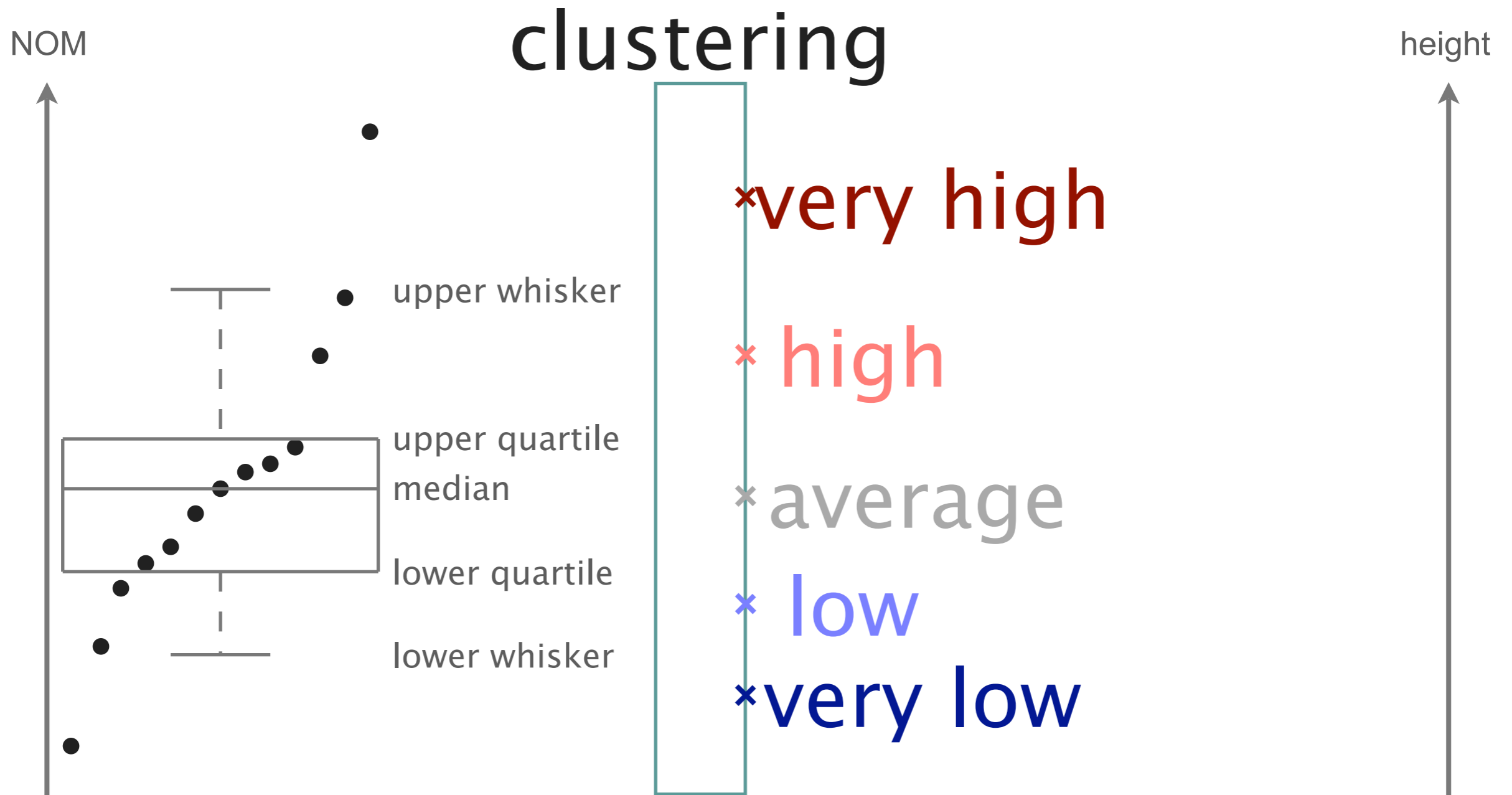
3. Boxplot-based mapping



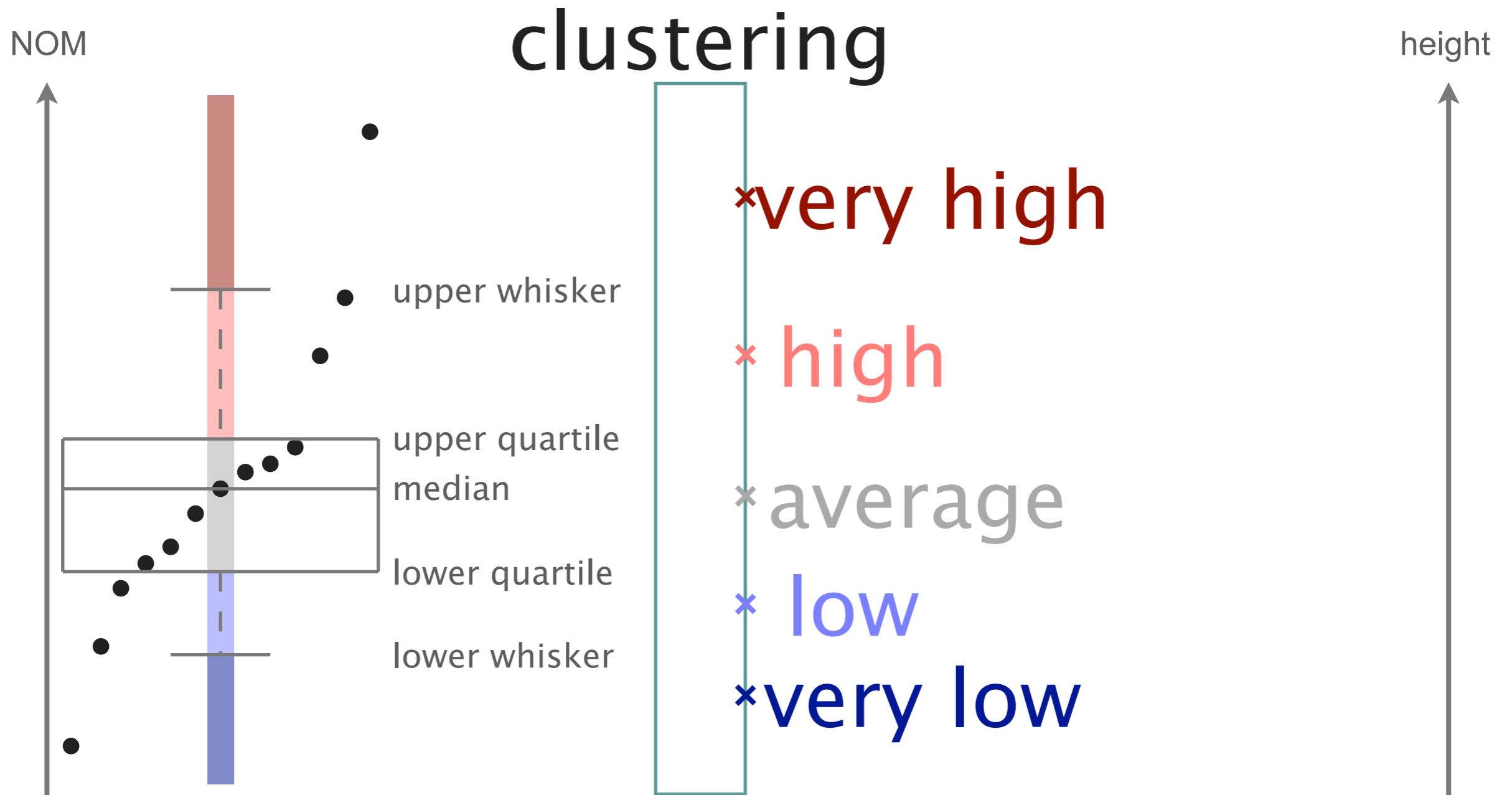
3. Boxplot-based mapping



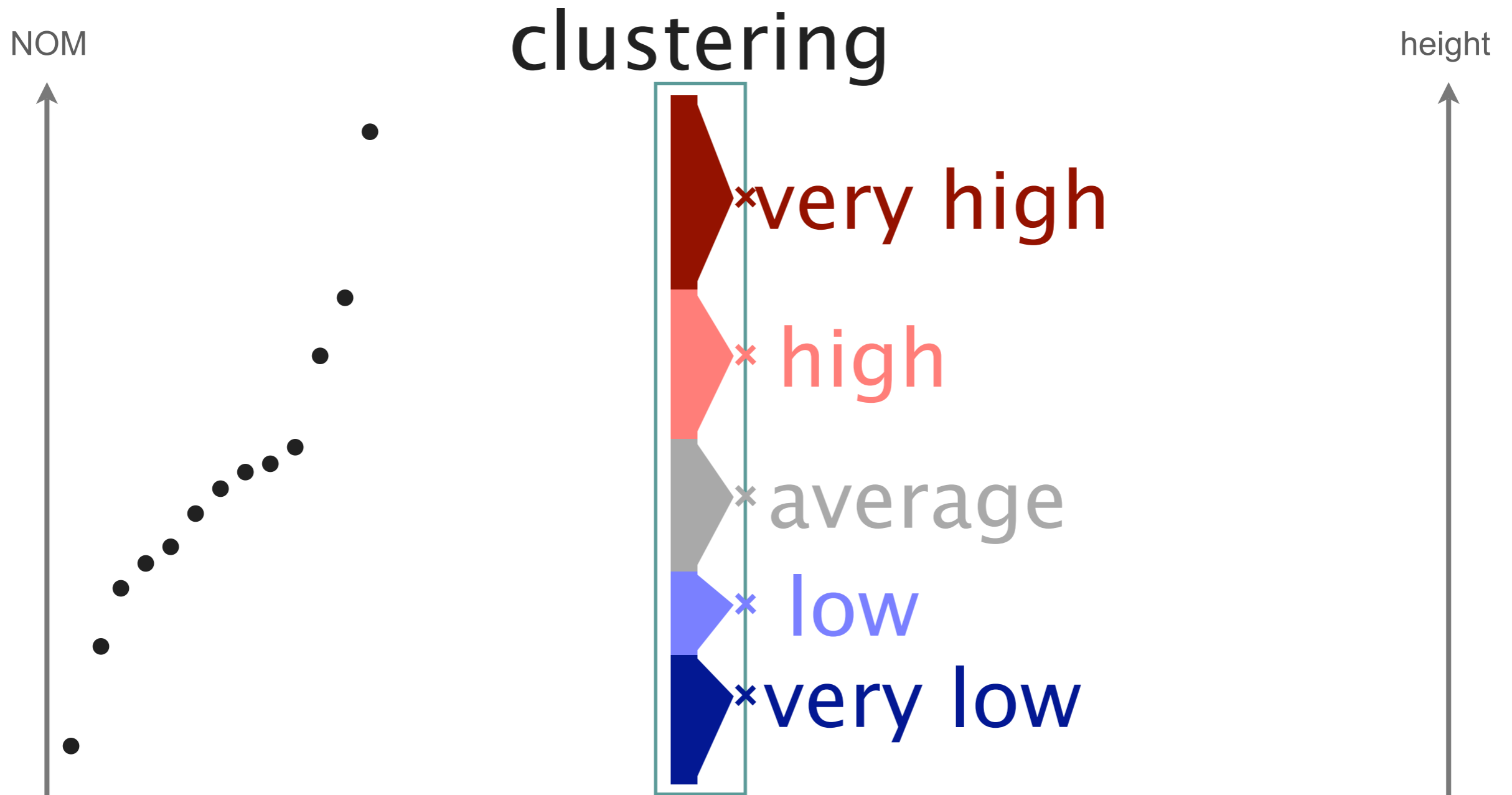
3. Boxplot-based mapping



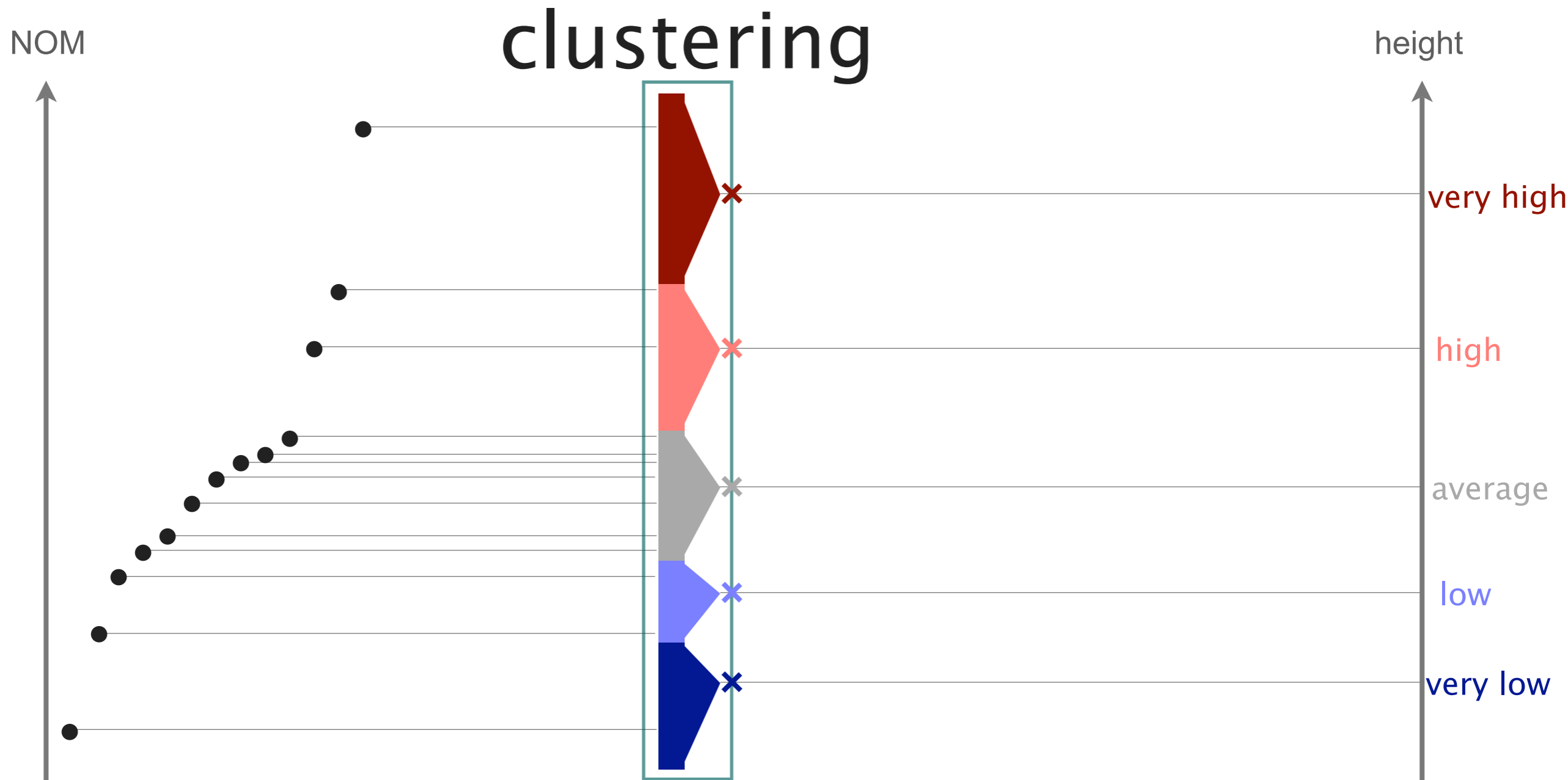
3. Boxplot-based mapping



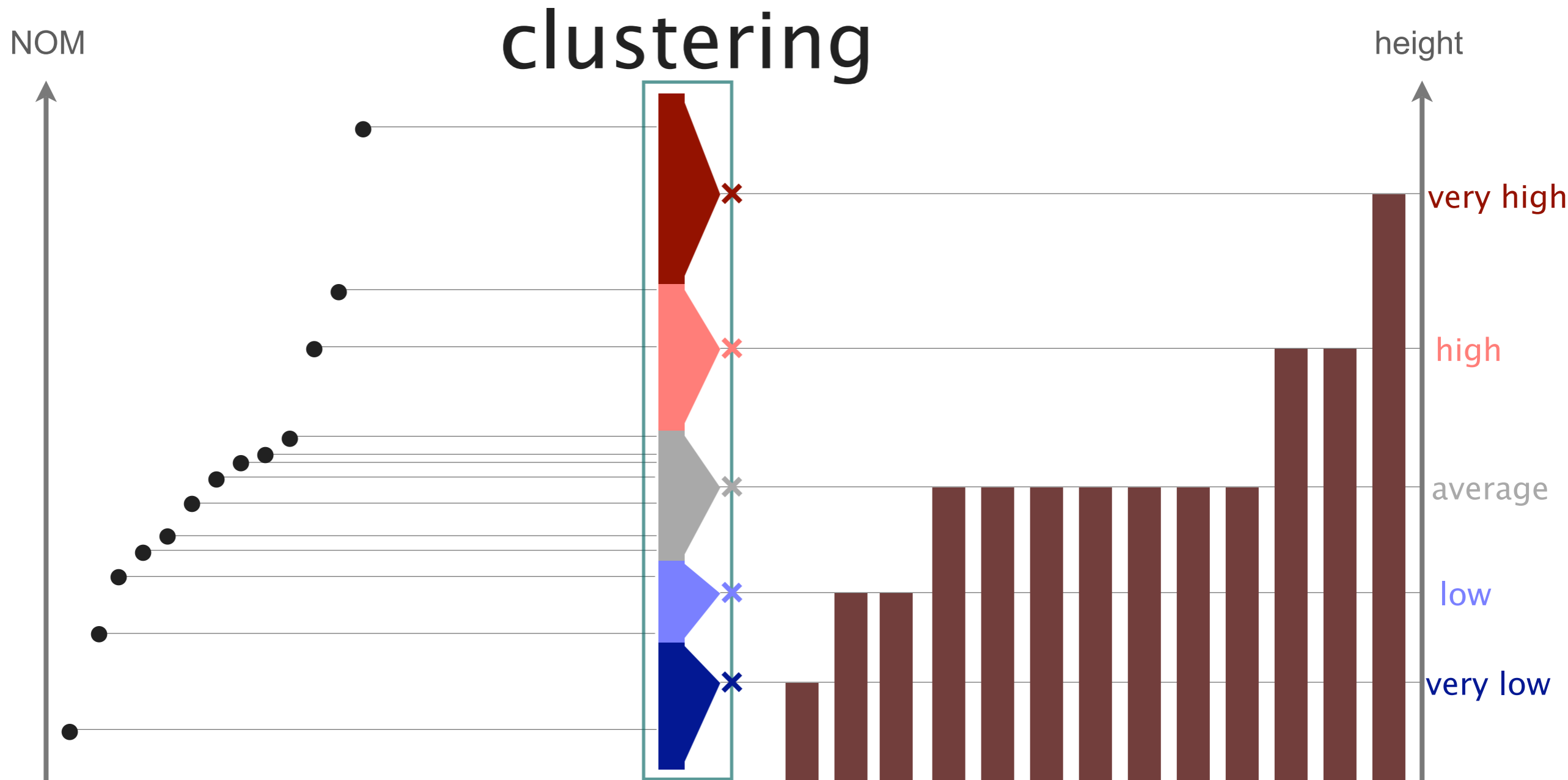
3. Boxplot-based mapping



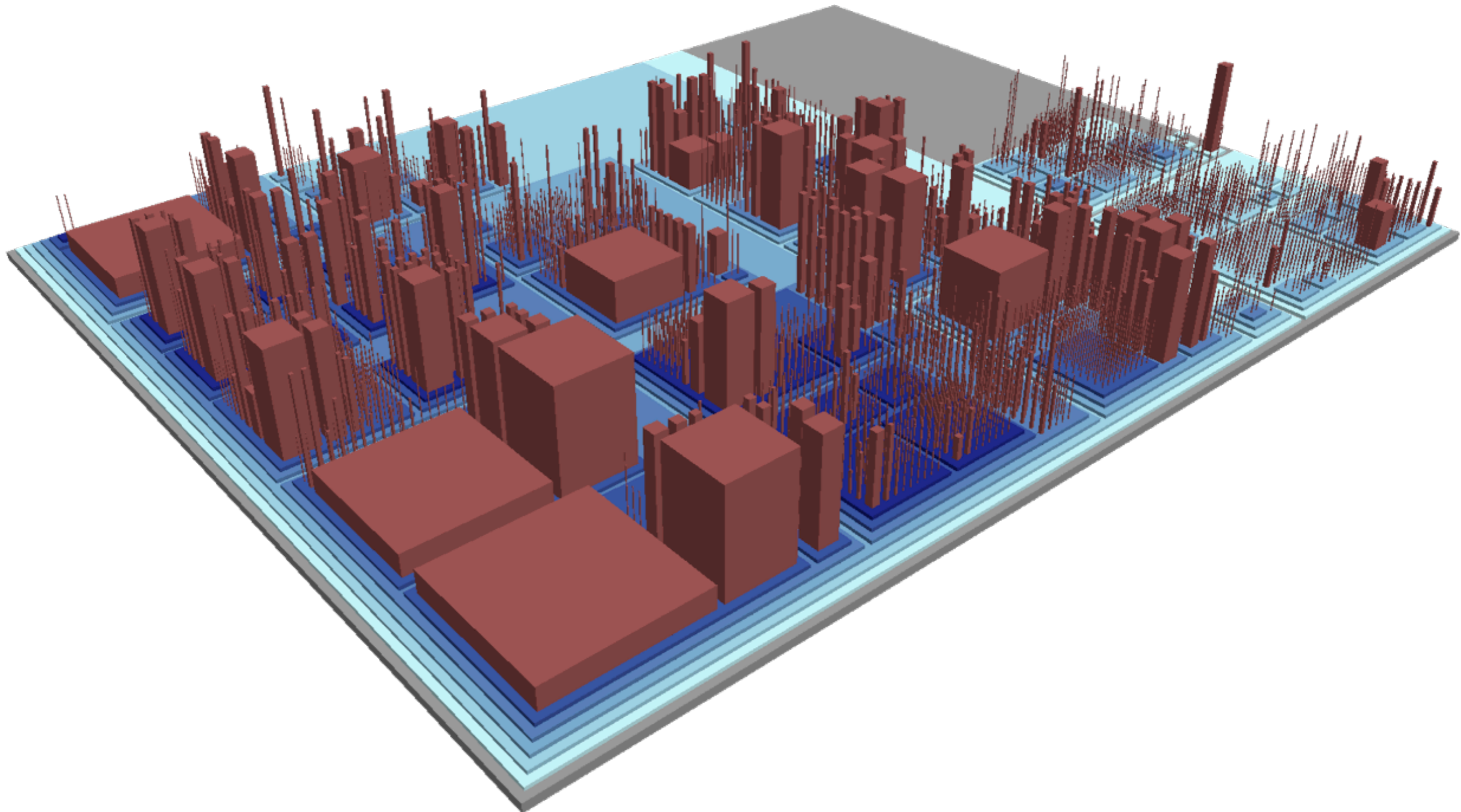
3. Boxplot-based mapping



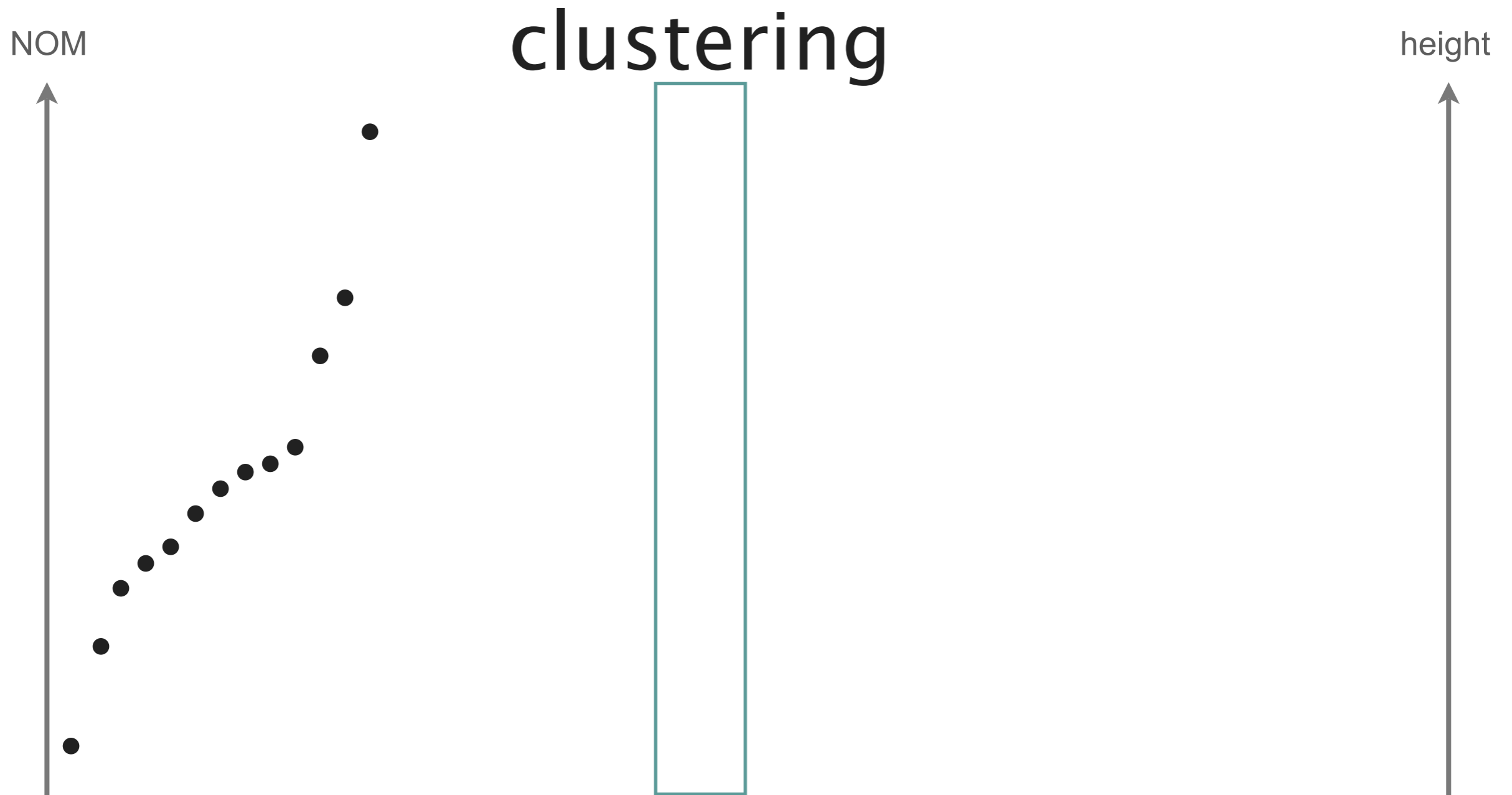
3. Boxplot-based mapping



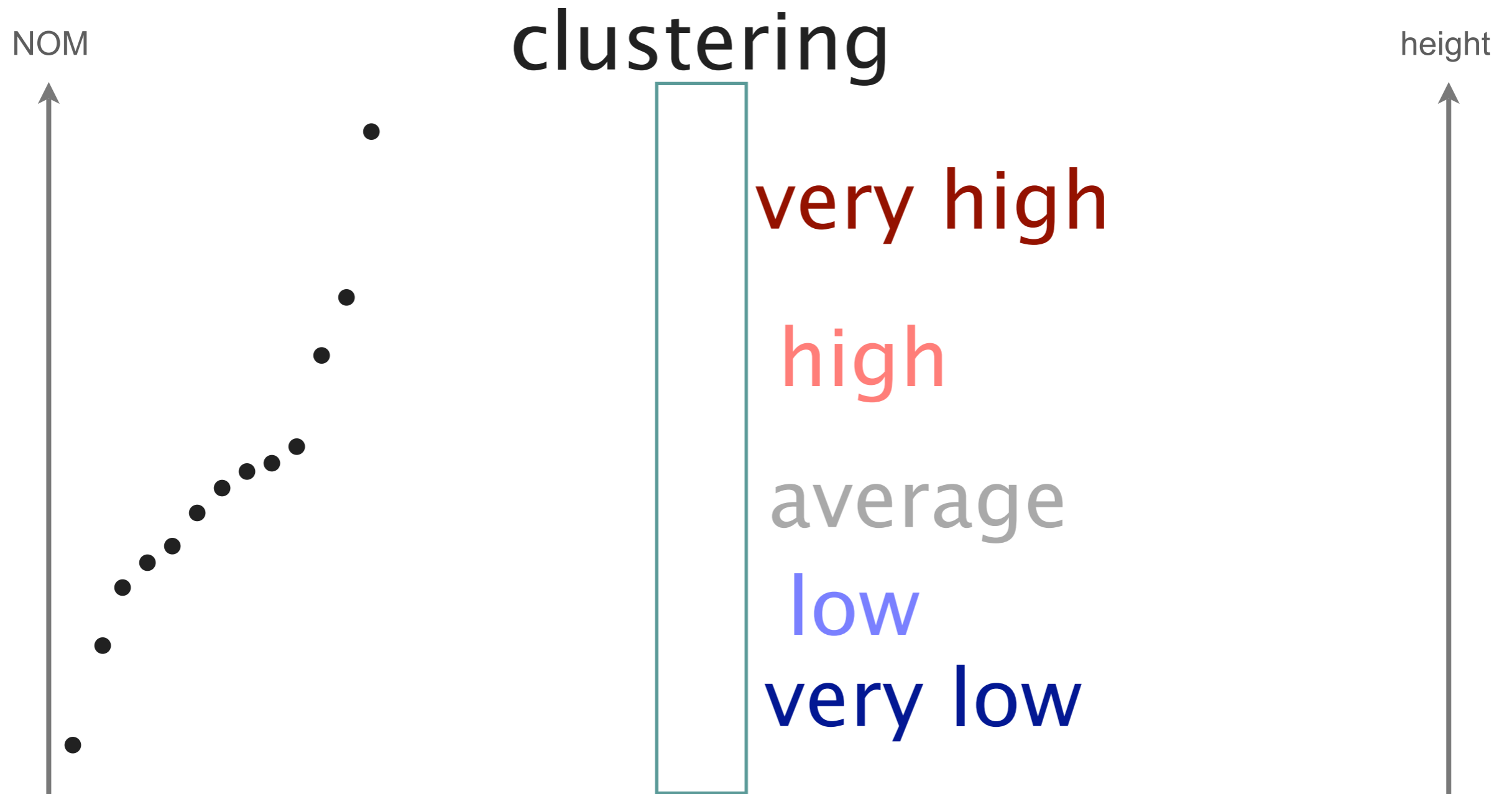
Boxplot-based mapping applied



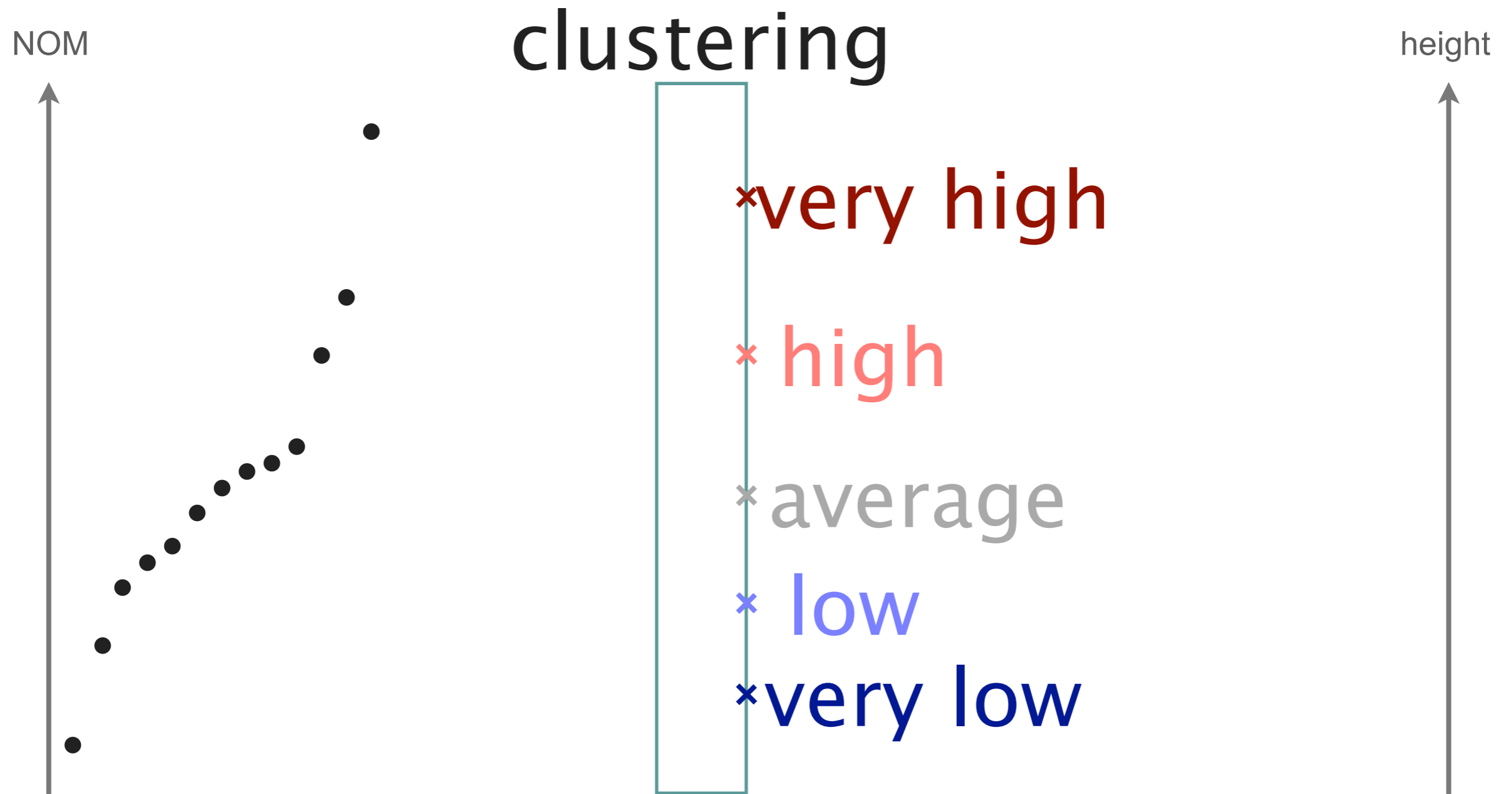
4. Threshold-based mapping



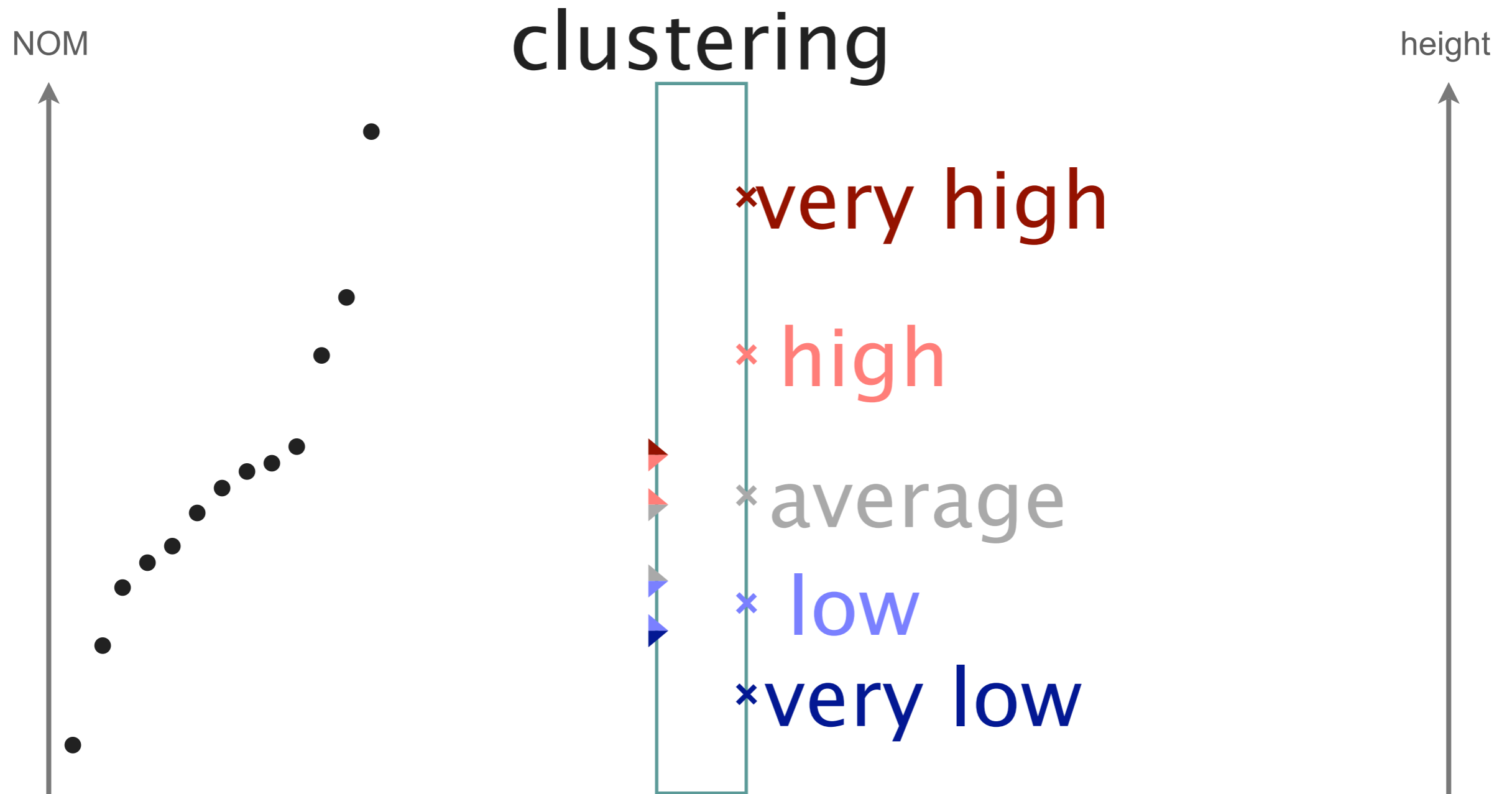
4. Threshold-based mapping



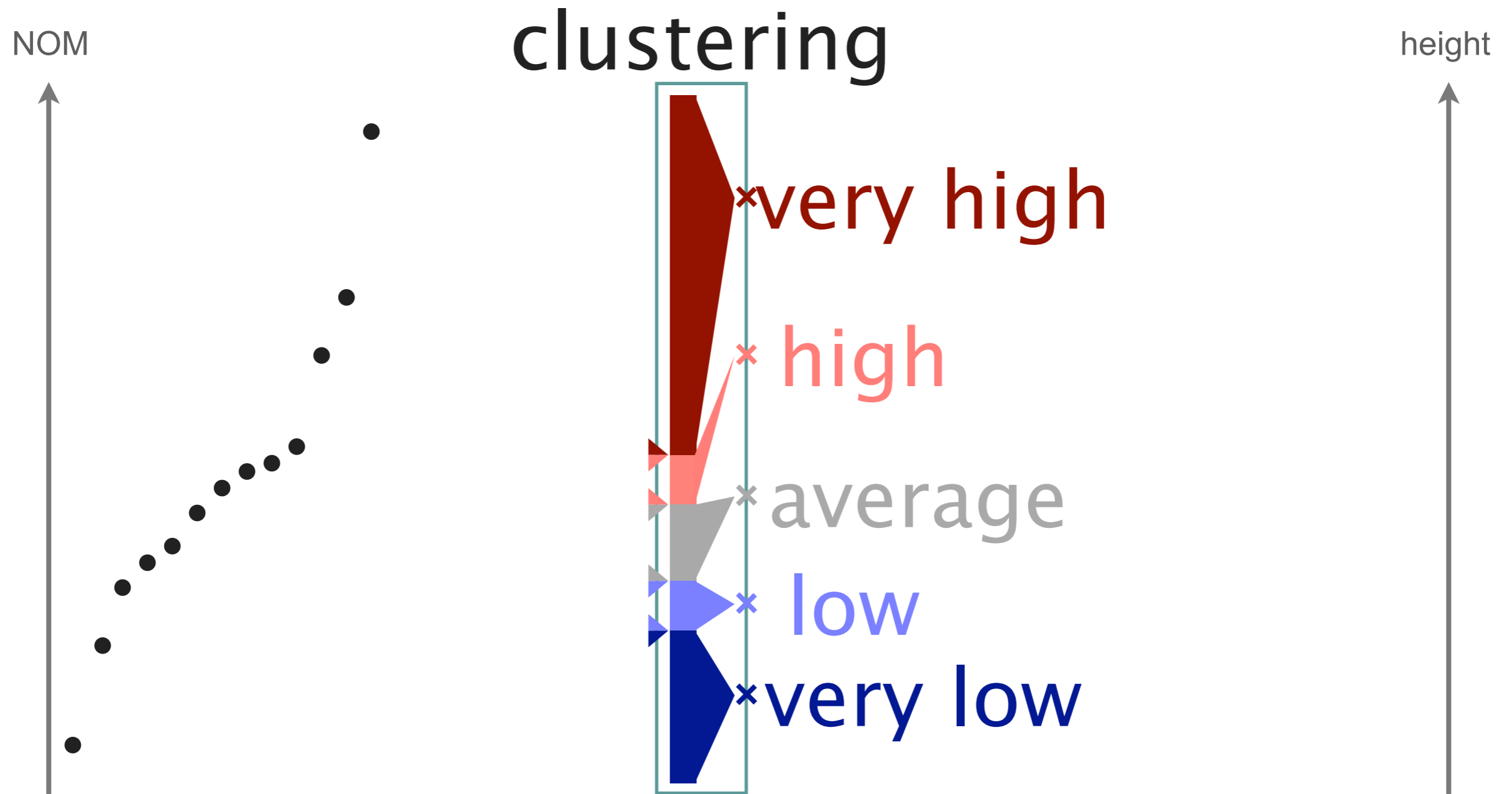
4. Threshold-based mapping



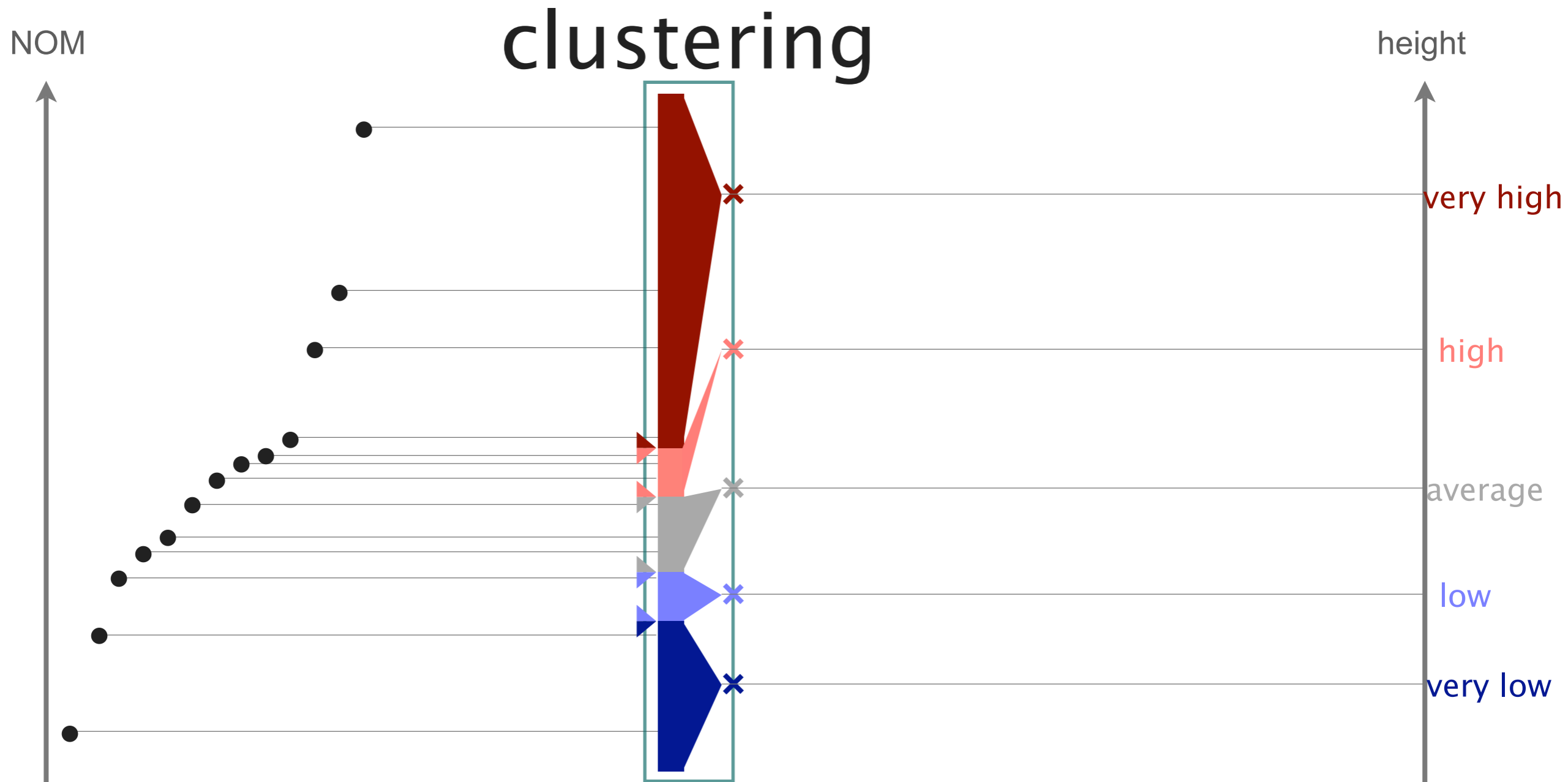
4. Threshold-based mapping



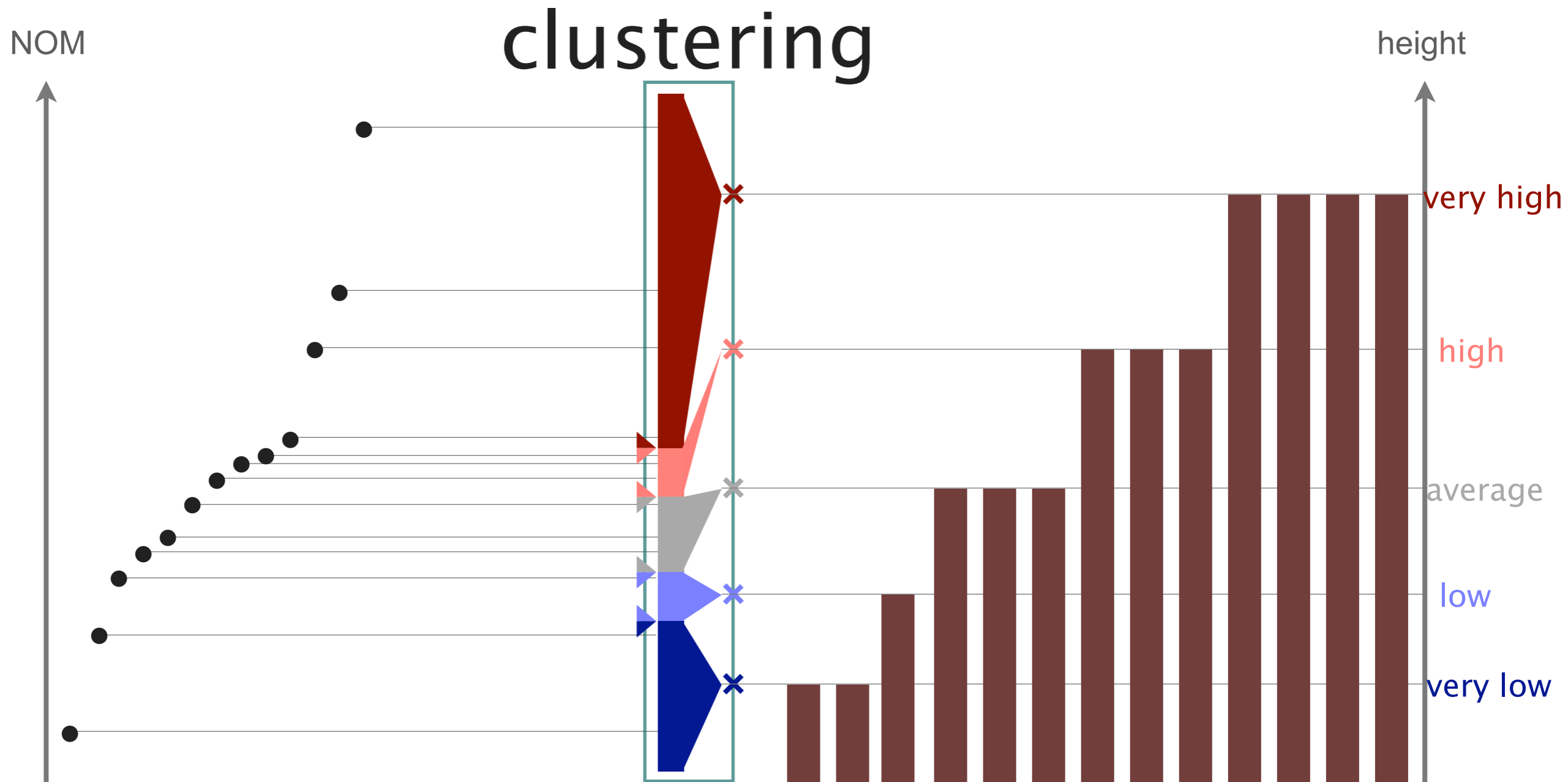
4. Threshold-based mapping



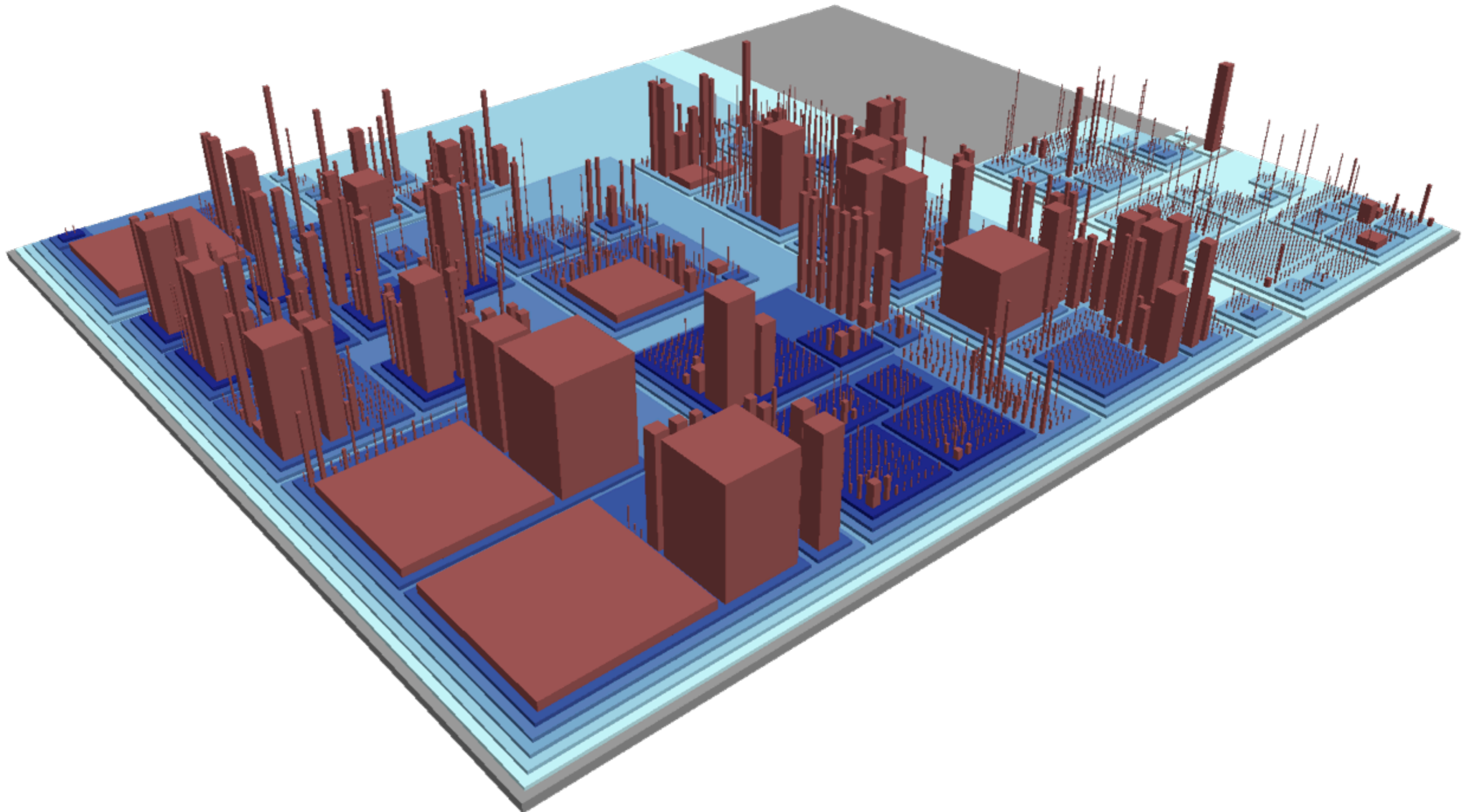
4. Threshold-based mapping



4. Threshold-based mapping



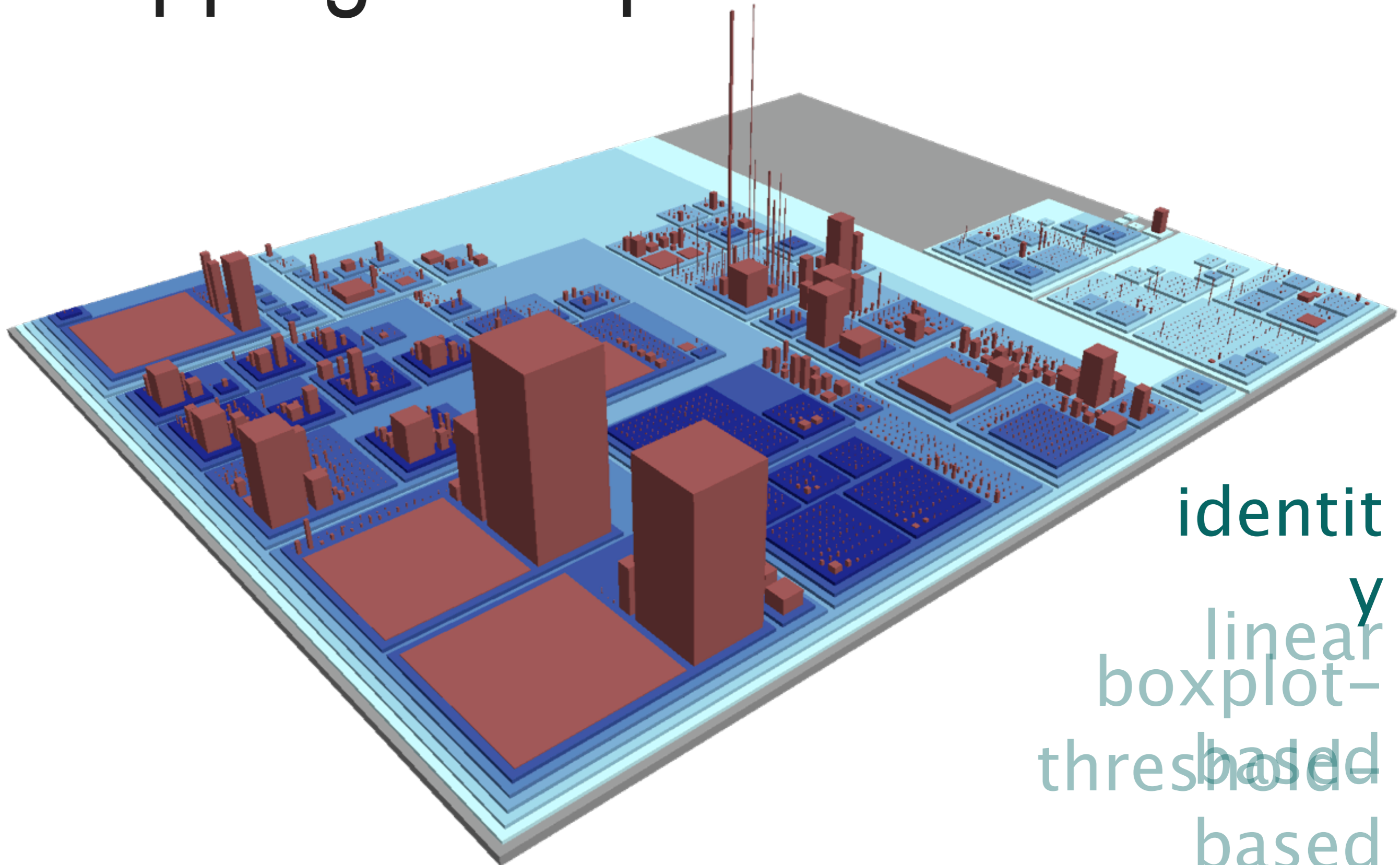
Threshold-based mapping applied



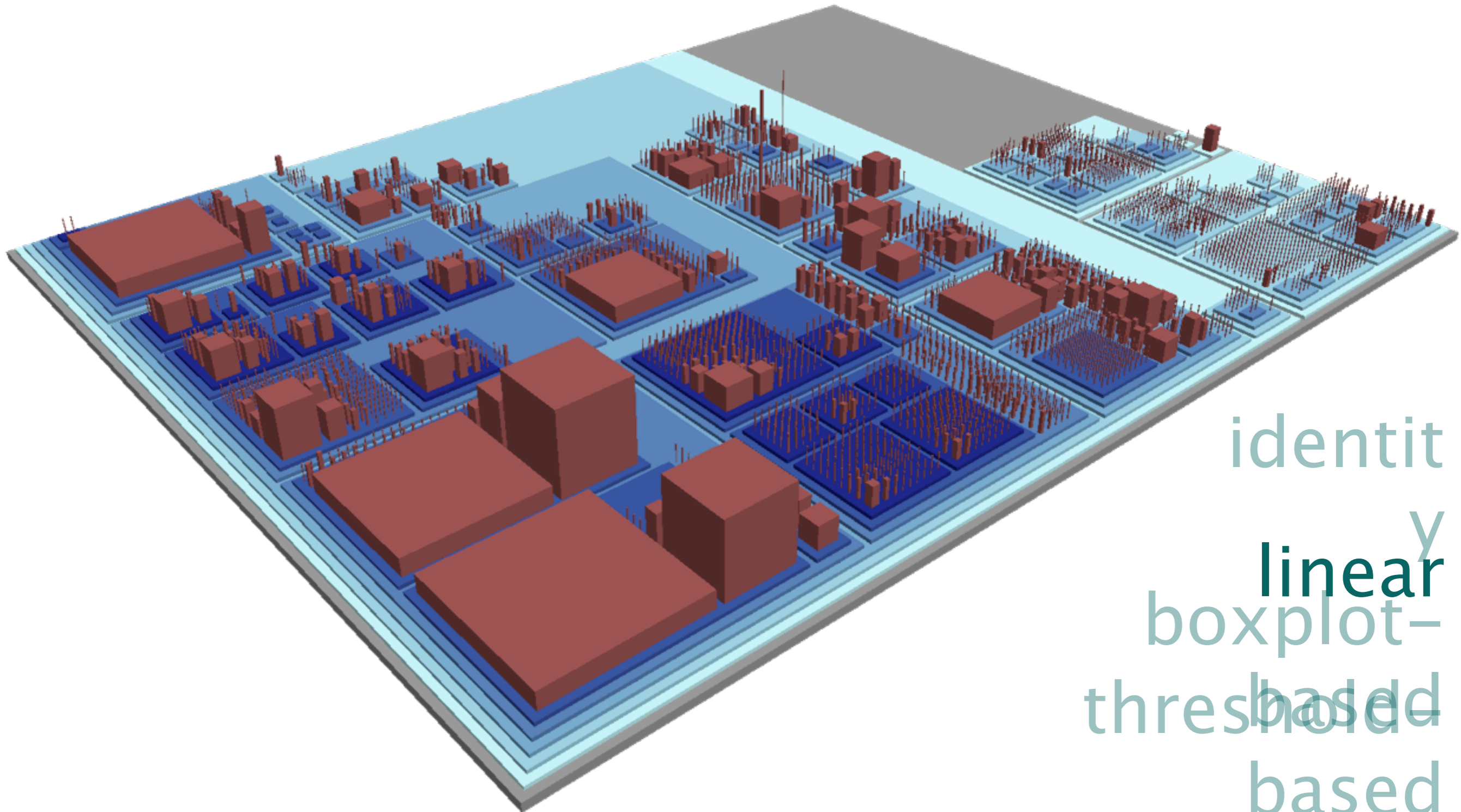
Mappings compared

identity
linear
boxplot-
threshold-
based
based

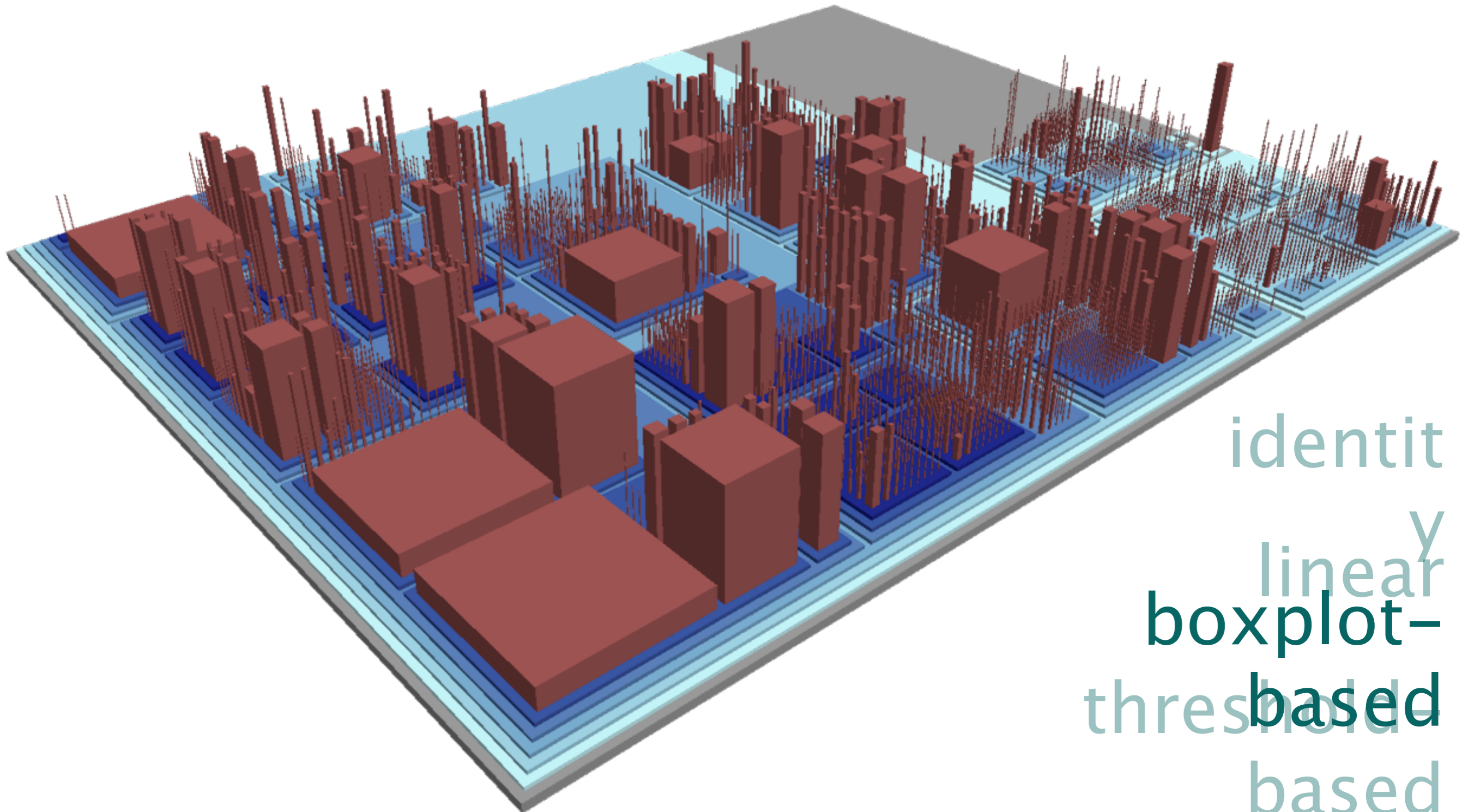
Mappings compared



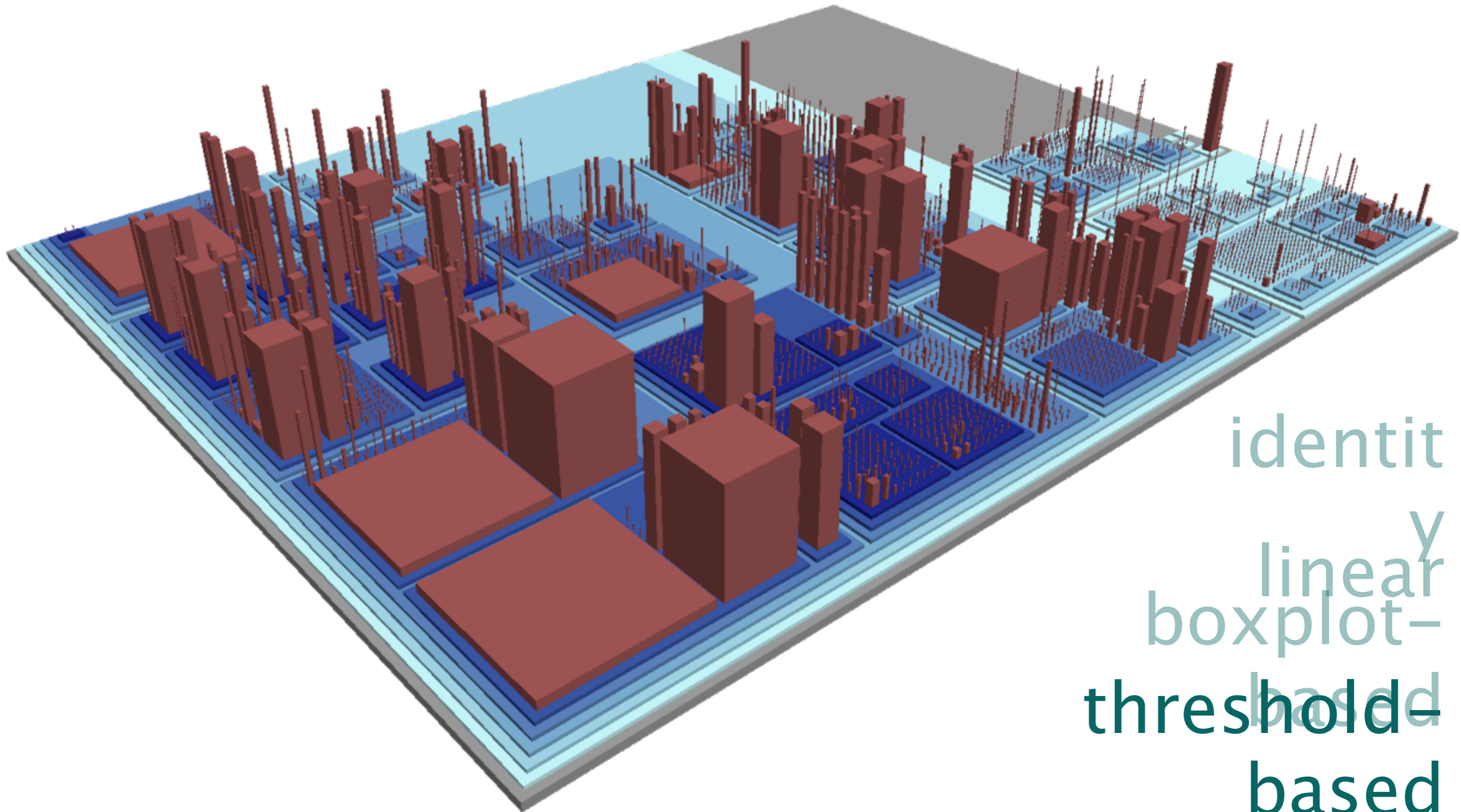
Mappings compared

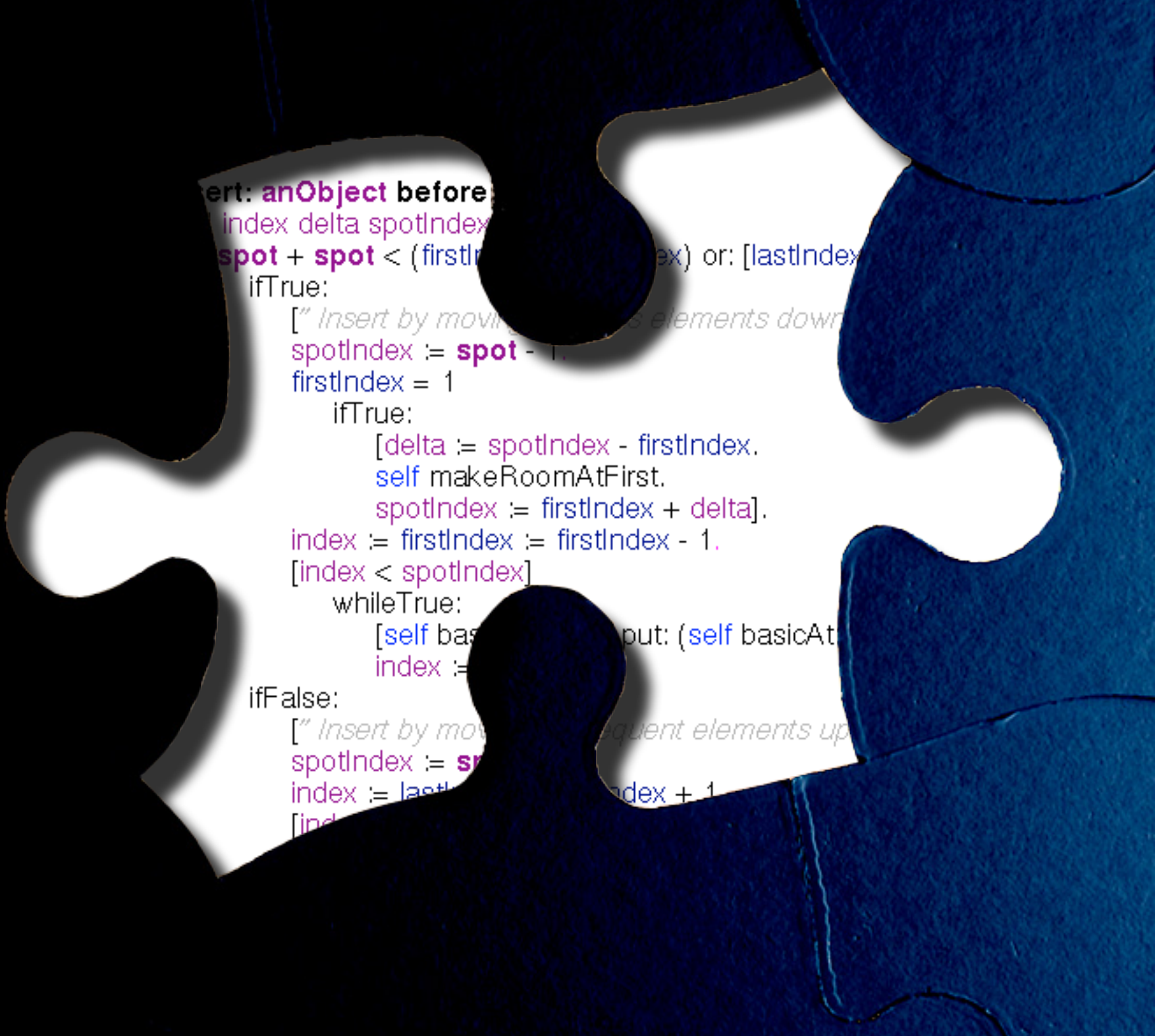


Mappings compared



Mappings compared

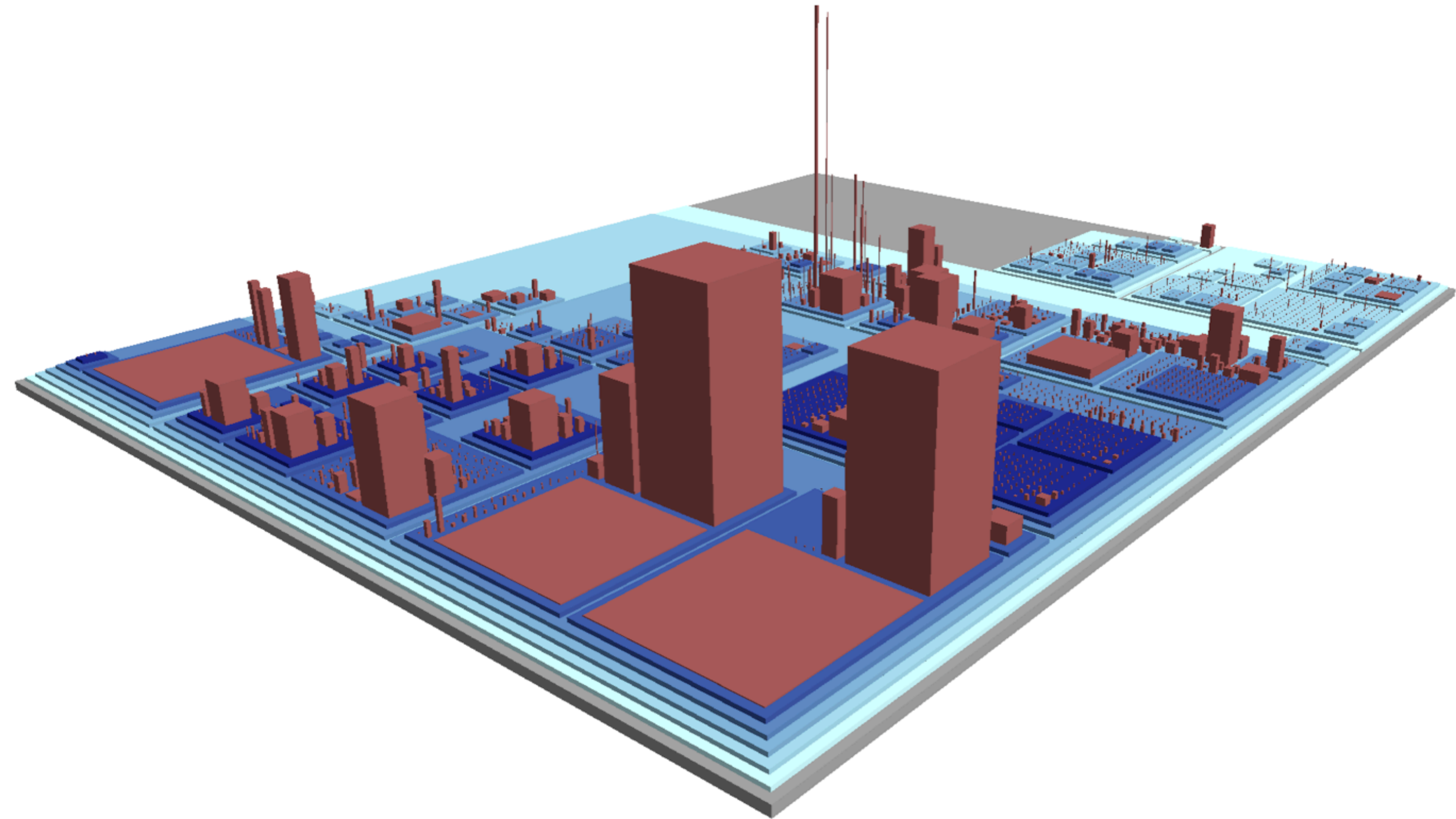




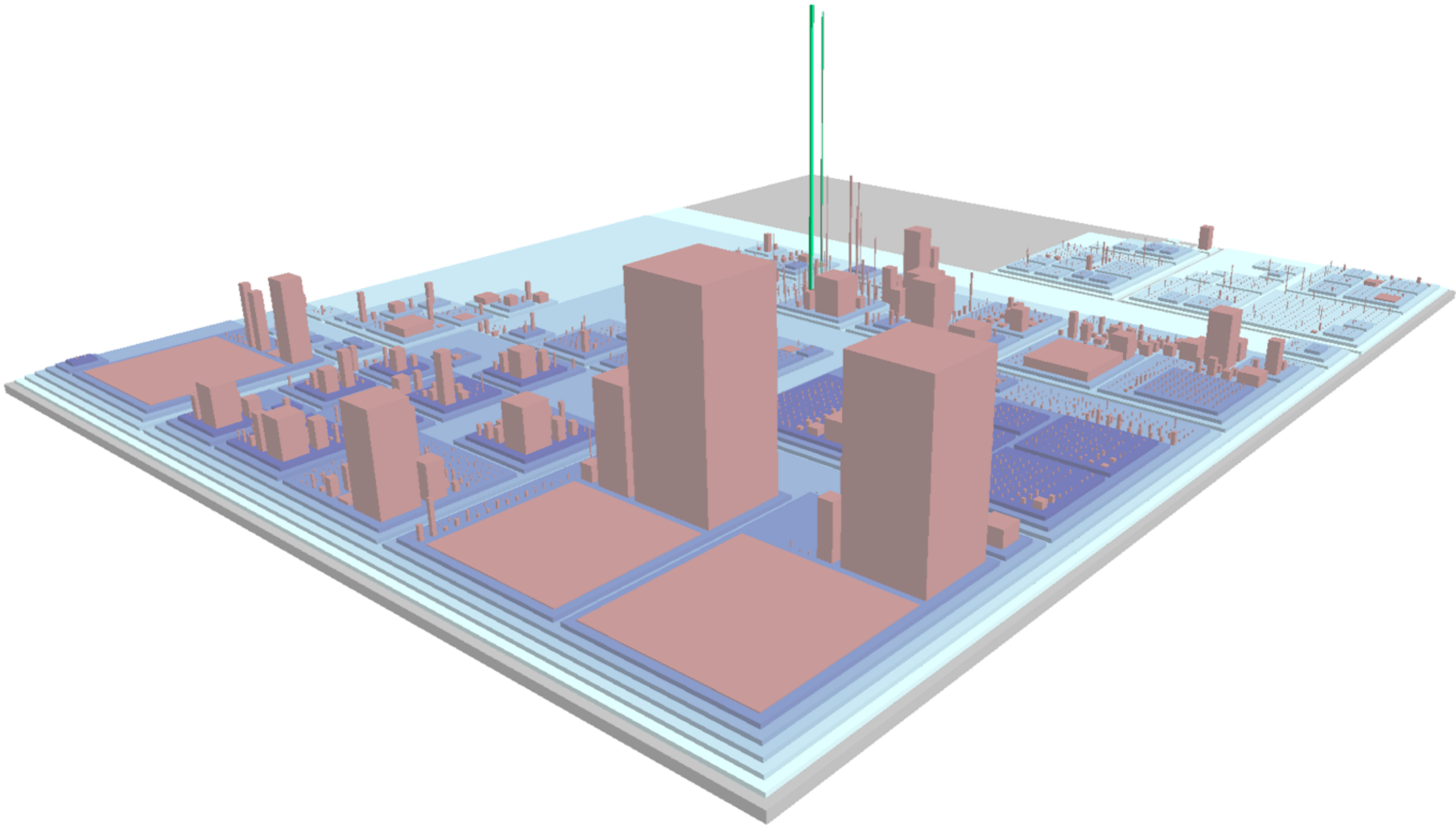
```
ert: anObject before
index delta spotIndex
spot + spot < (firstIndex or: [lastIndex
ifTrue:
  [" Insert by moving subsequent elements down
spotIndex := spot - 1.
firstIndex = 1
ifTrue:
  [delta := spotIndex - firstIndex.
self makeRoomAtFirst.
spotIndex := firstIndex + delta].
index := firstIndex := firstIndex - 1.
[index < spotIndex]
whileTrue:
  [self basicAt: index put: (self basicAt:
index :=
ifFalse:
  [" Insert by moving subsequent elements up
spotIndex := spot
index := lastIndex := lastIndex + 1.
[index
```

Understanding programs

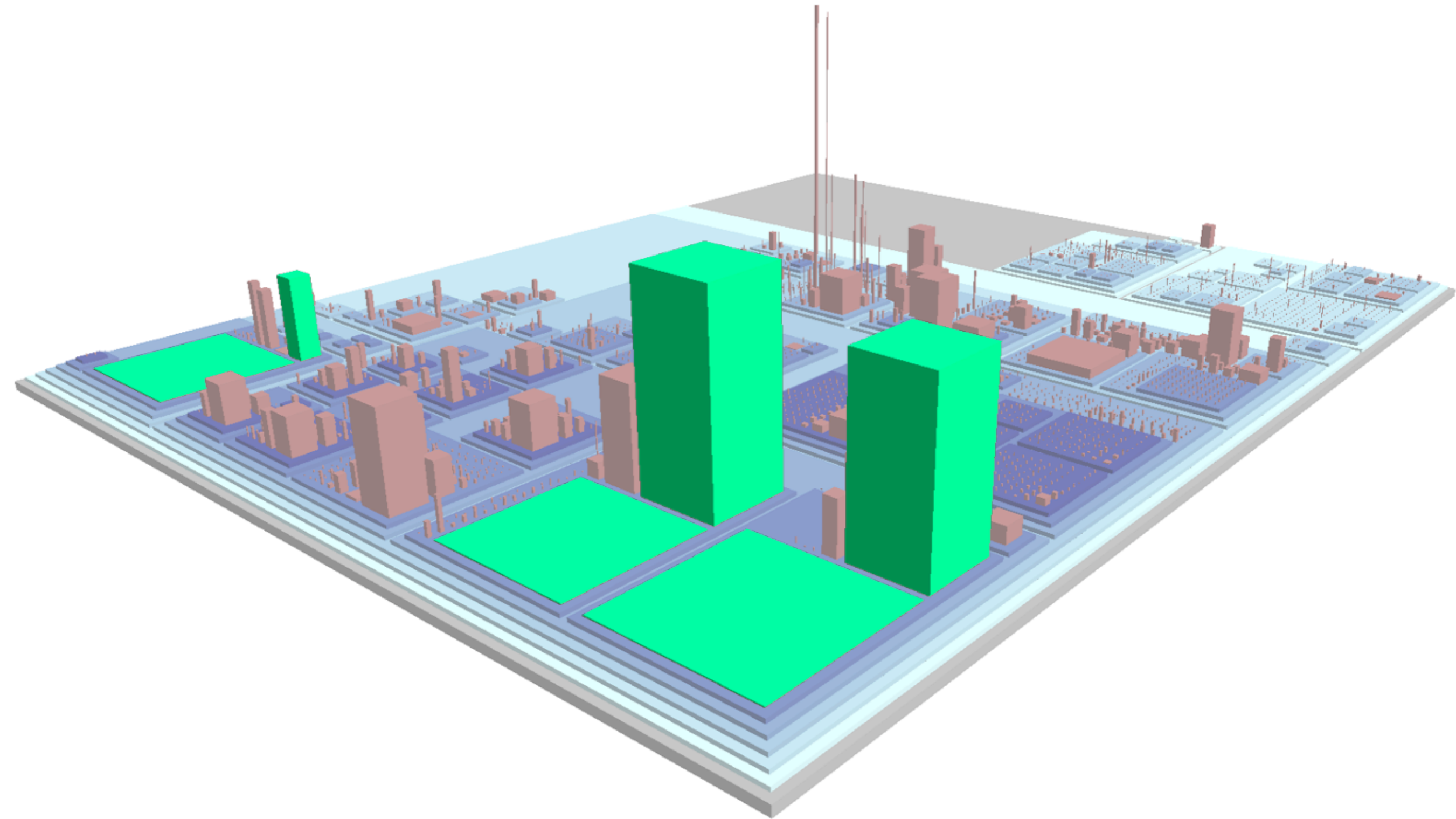
Reverse-engineering ArgoUML



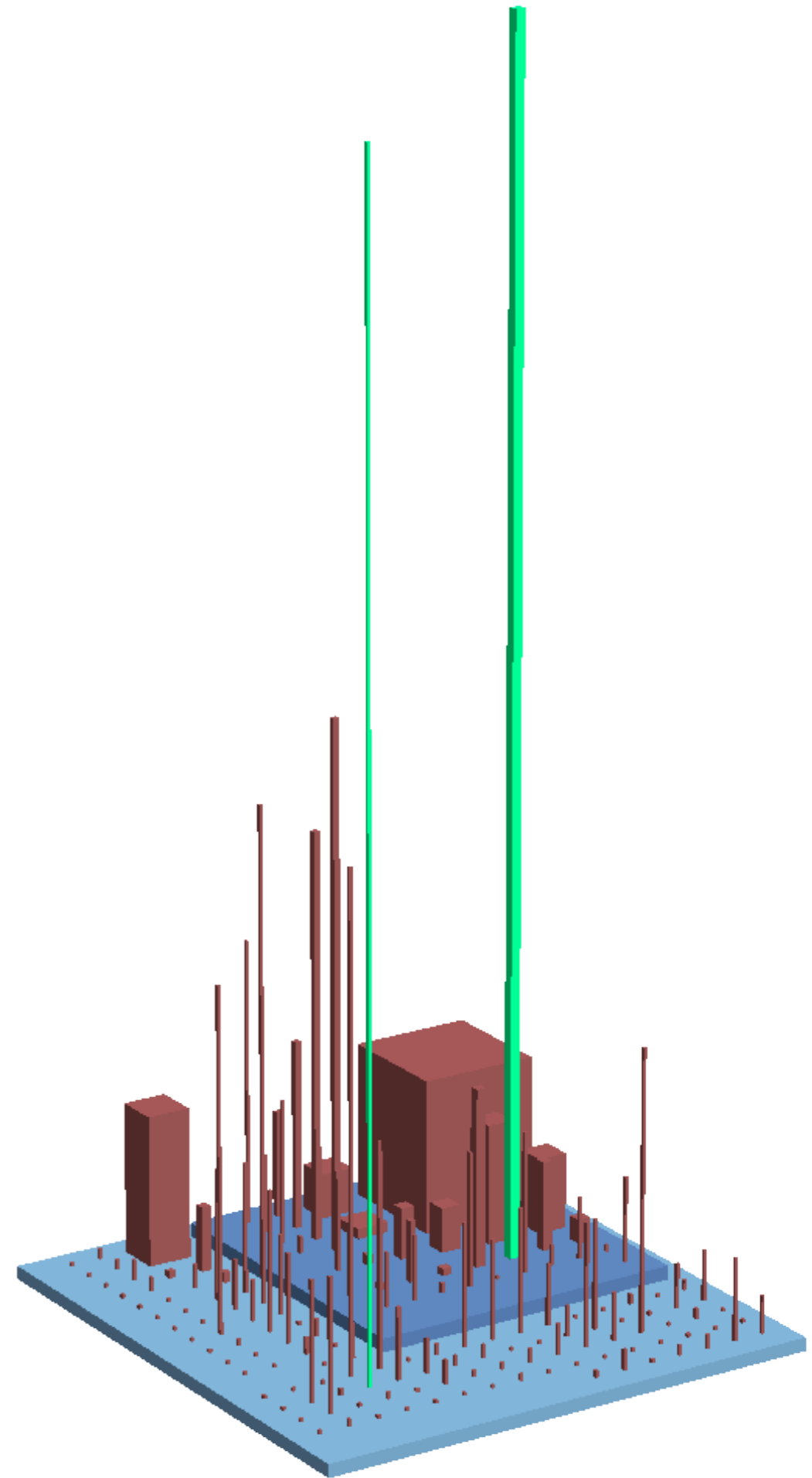
Reverse-engineering ArgoUML



Reverse-engineering ArgoUML

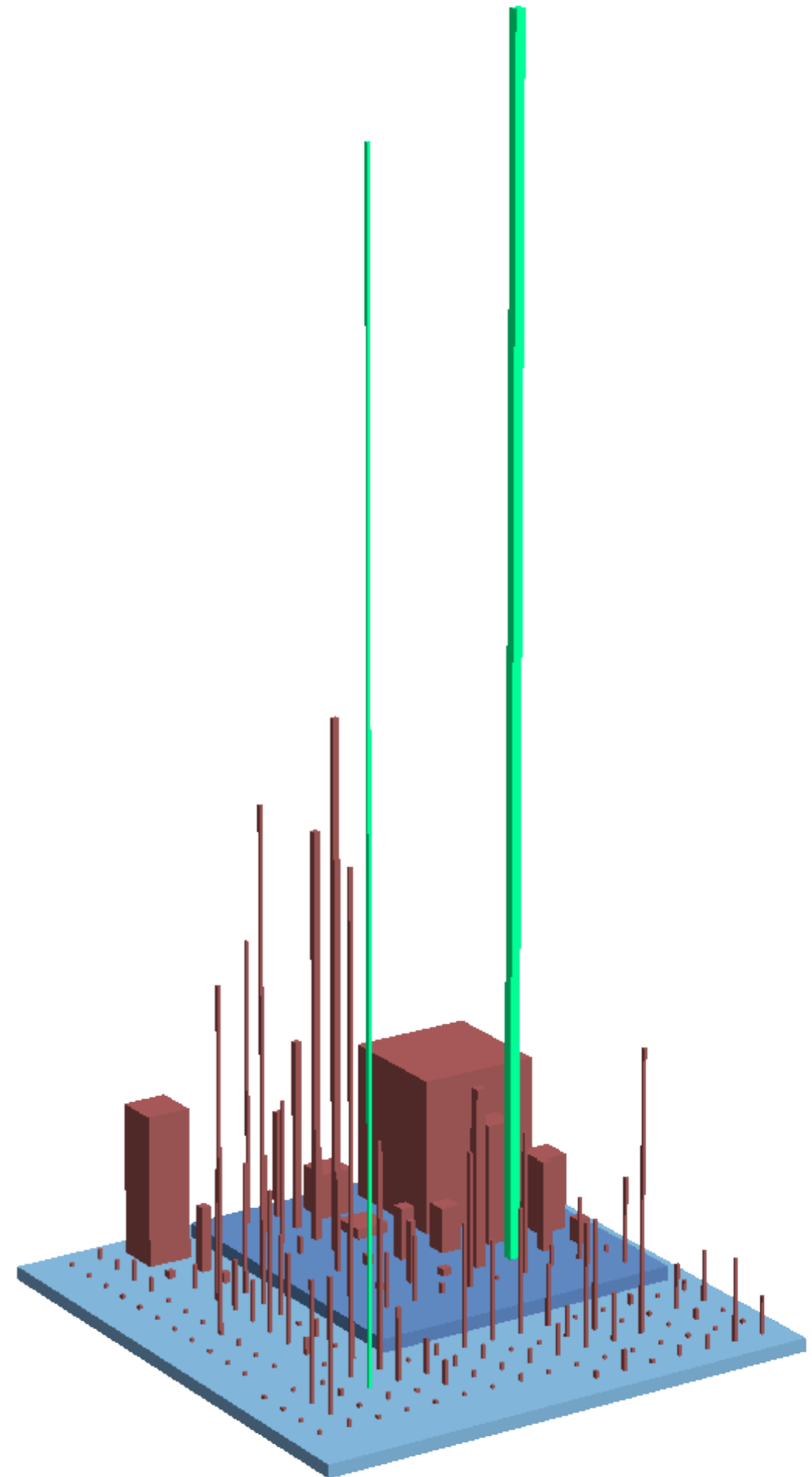


The twin towers

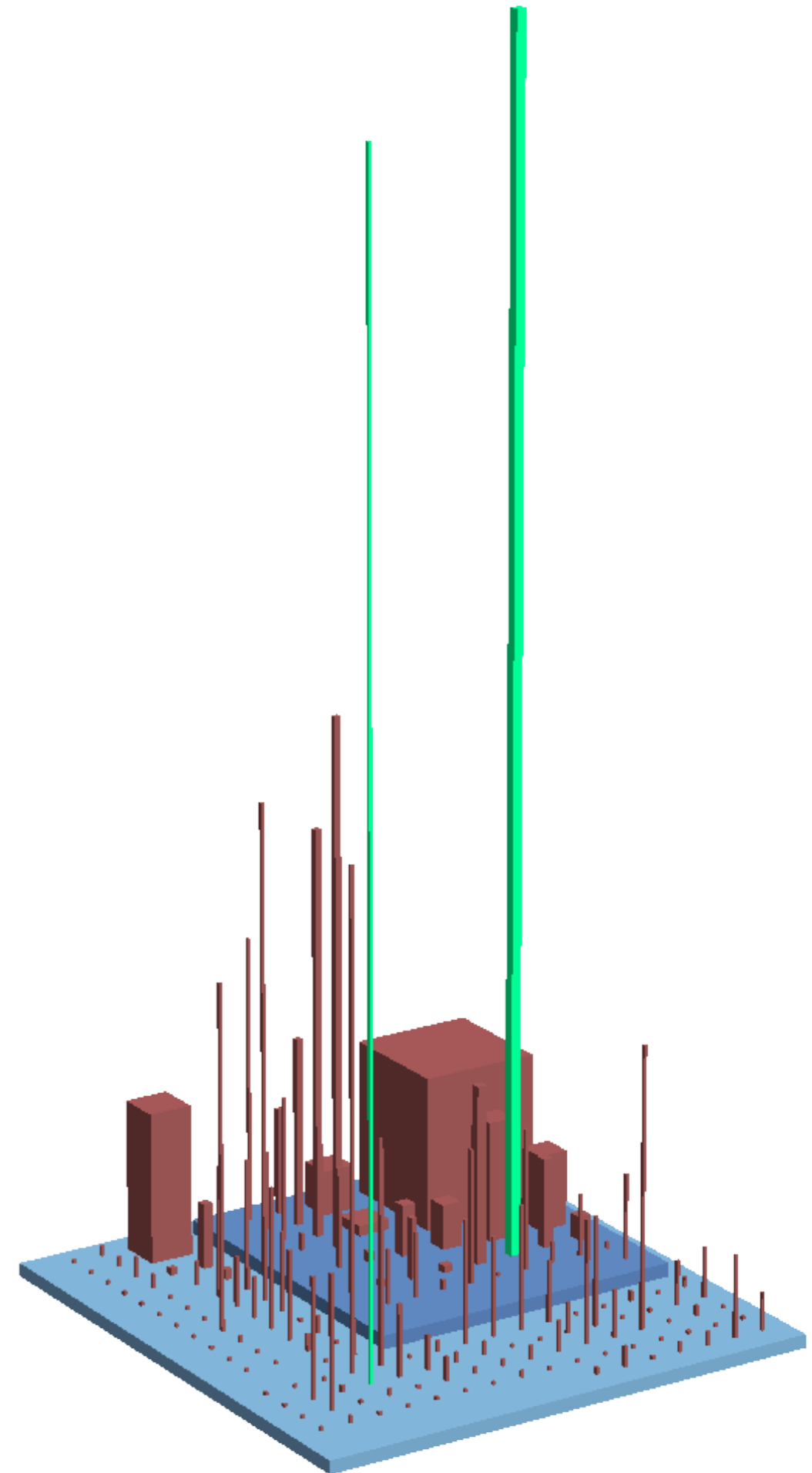
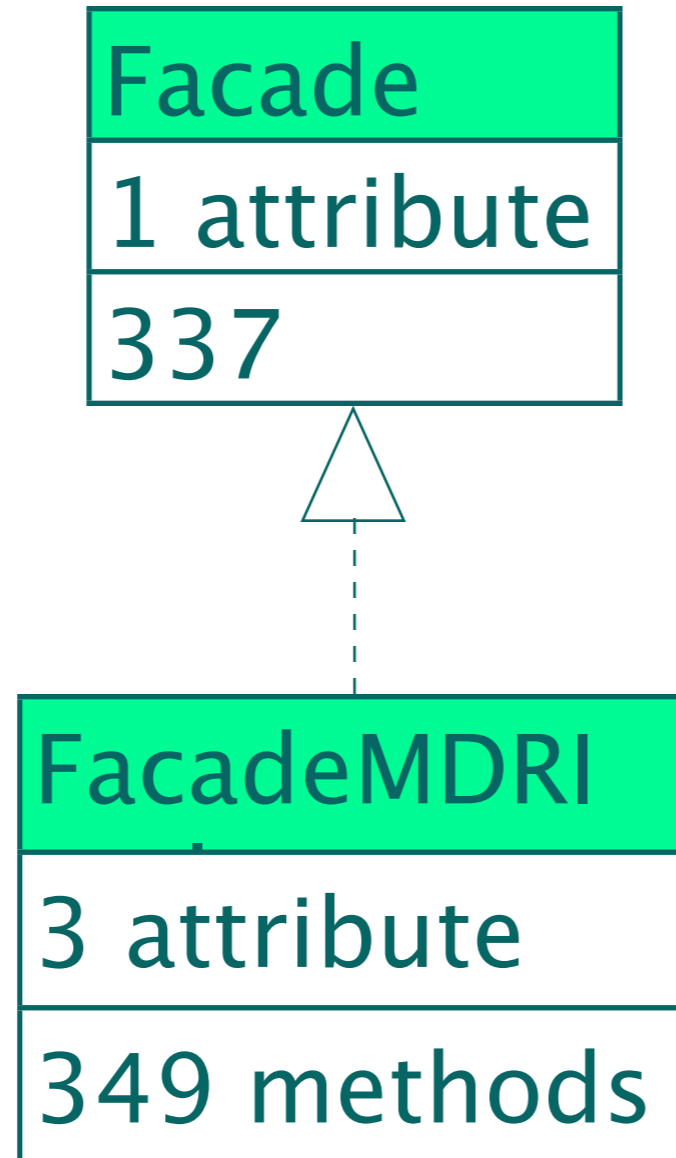


The twin towers

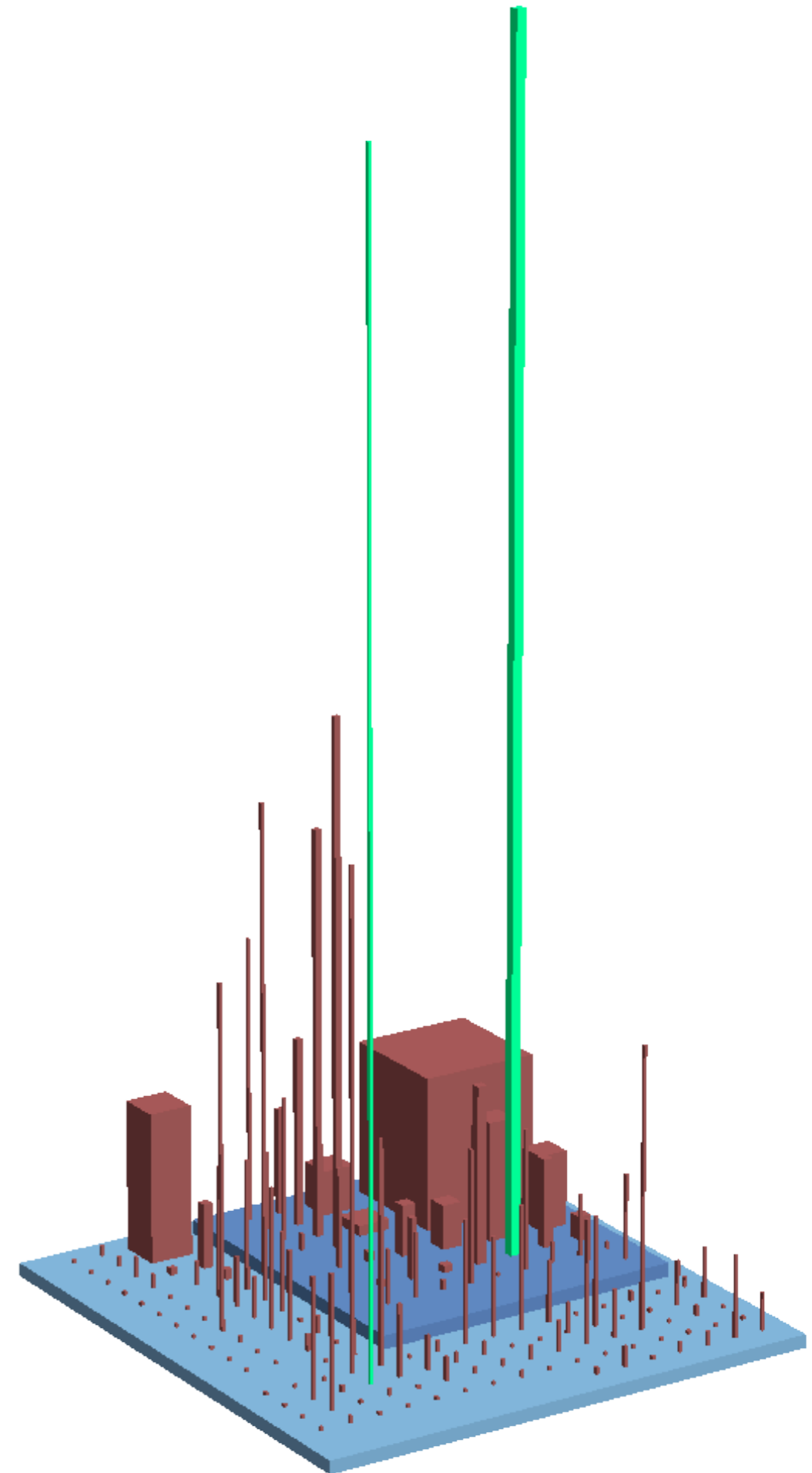
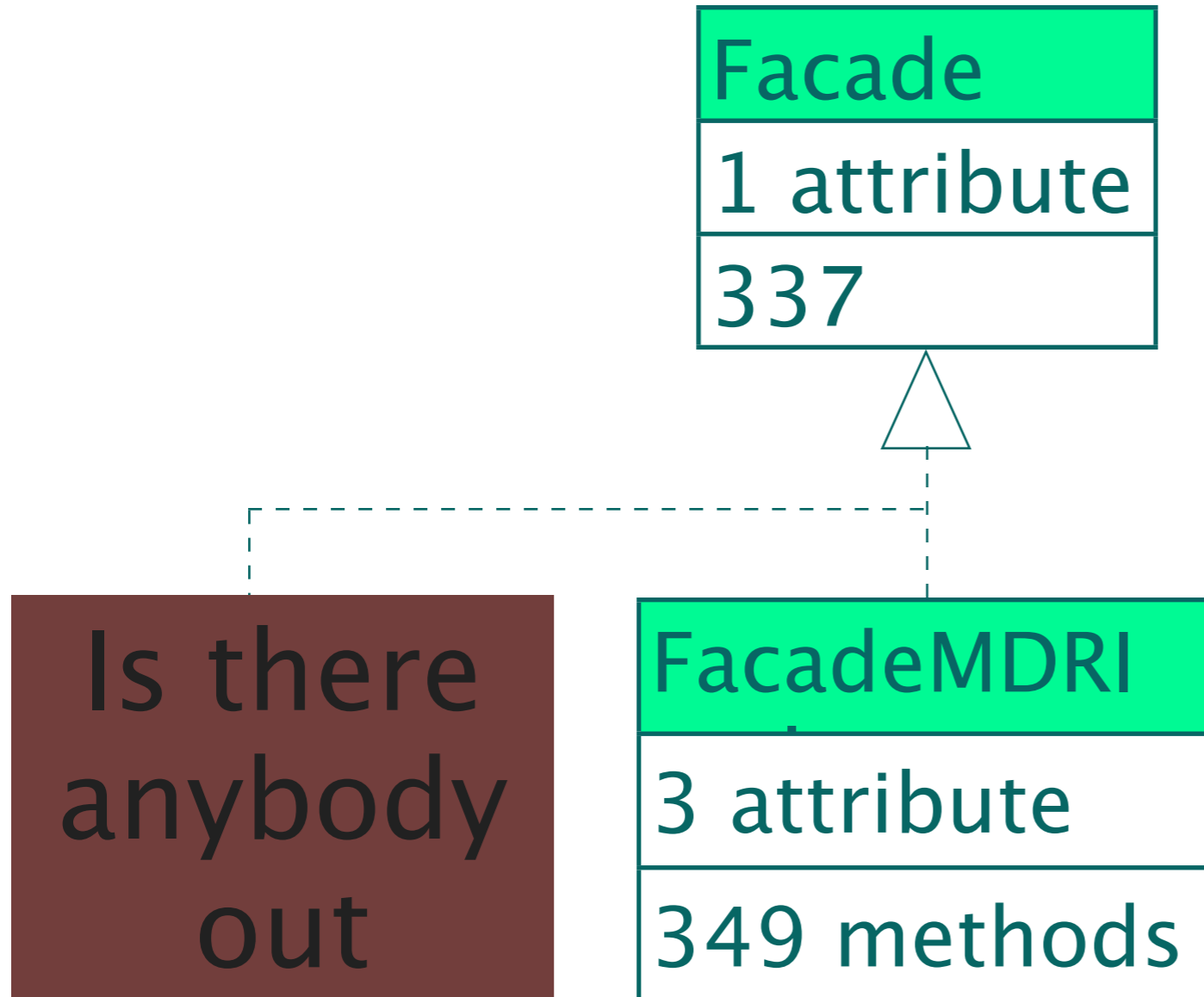
Facade
1 attribute
337



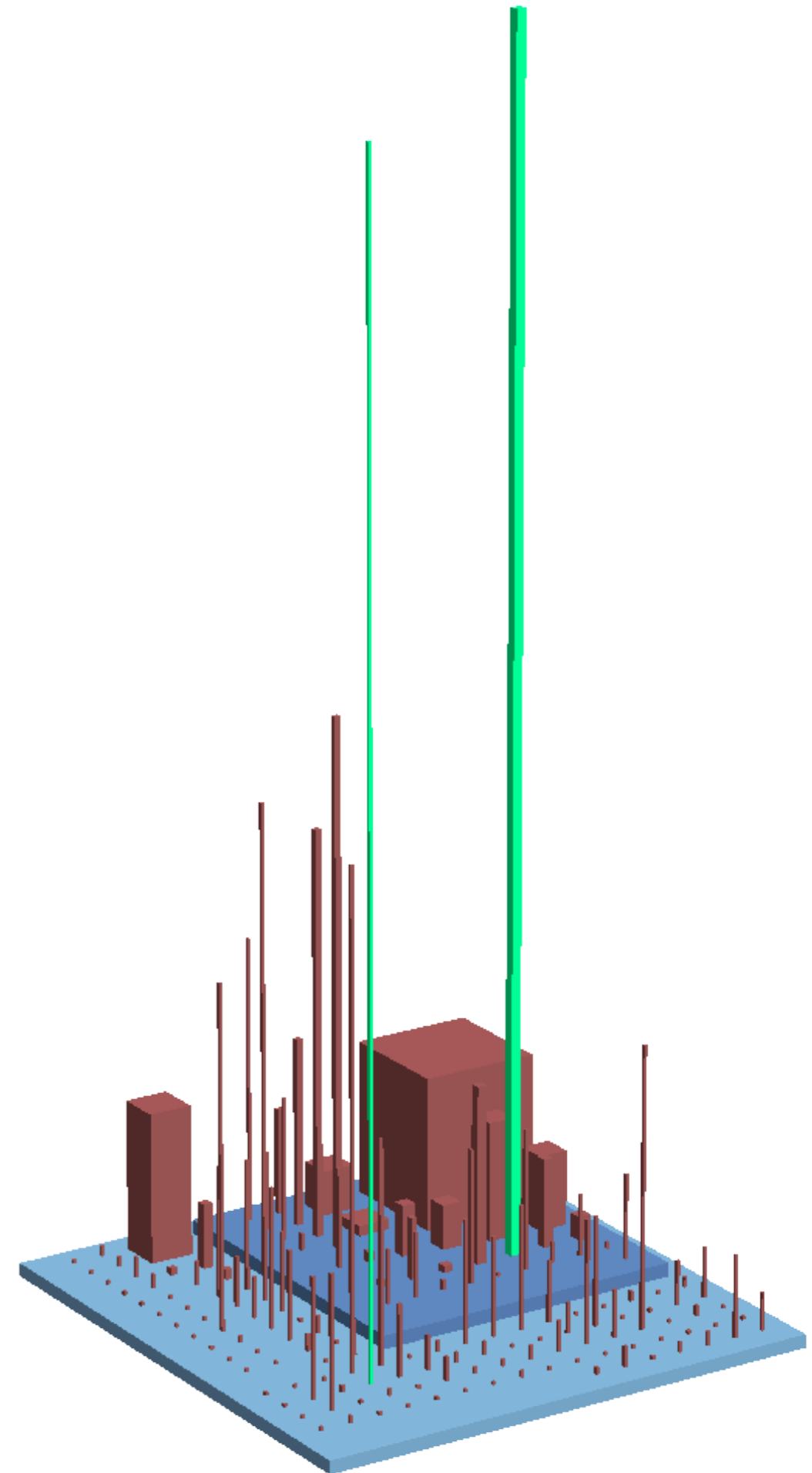
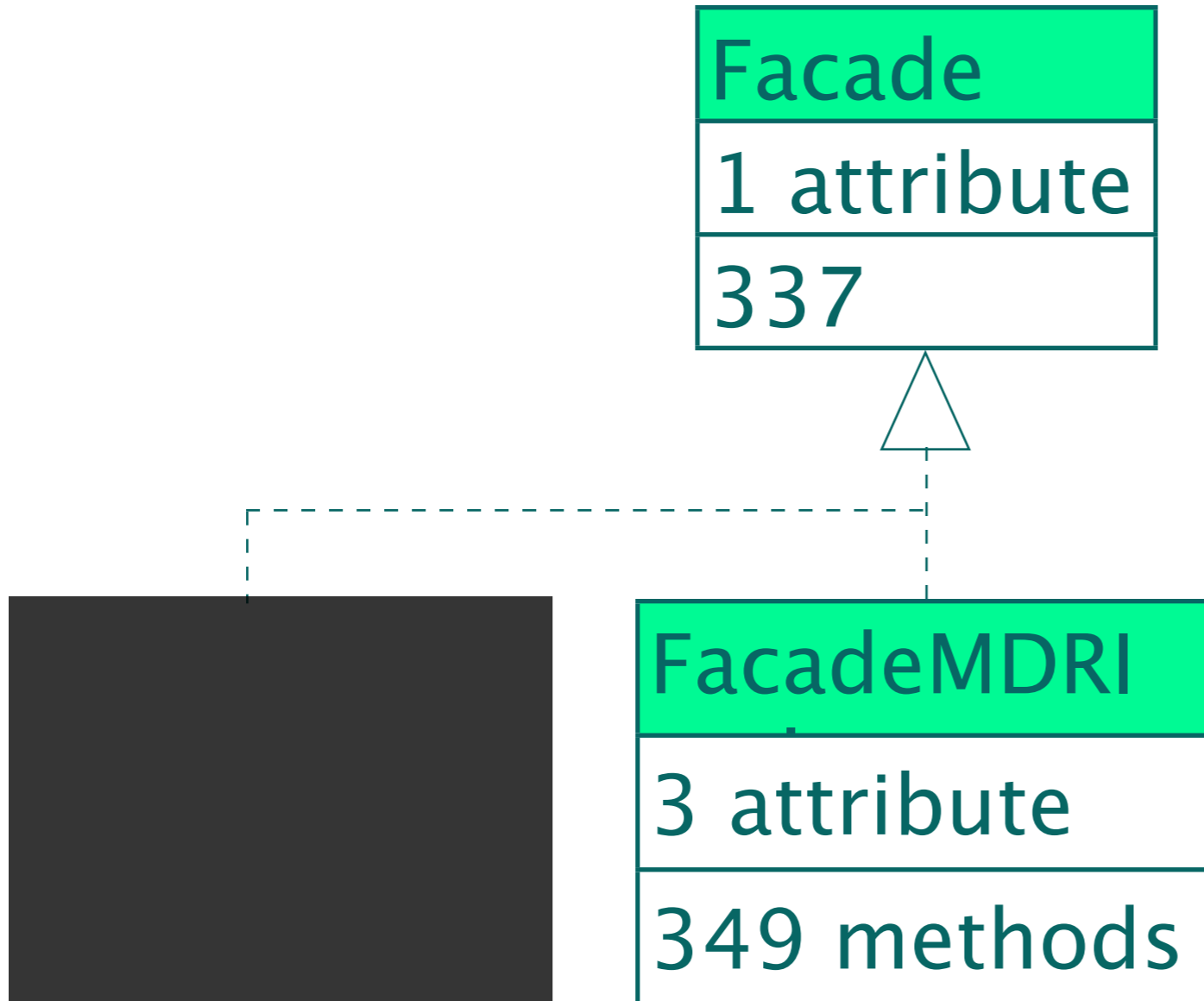
The twin towers



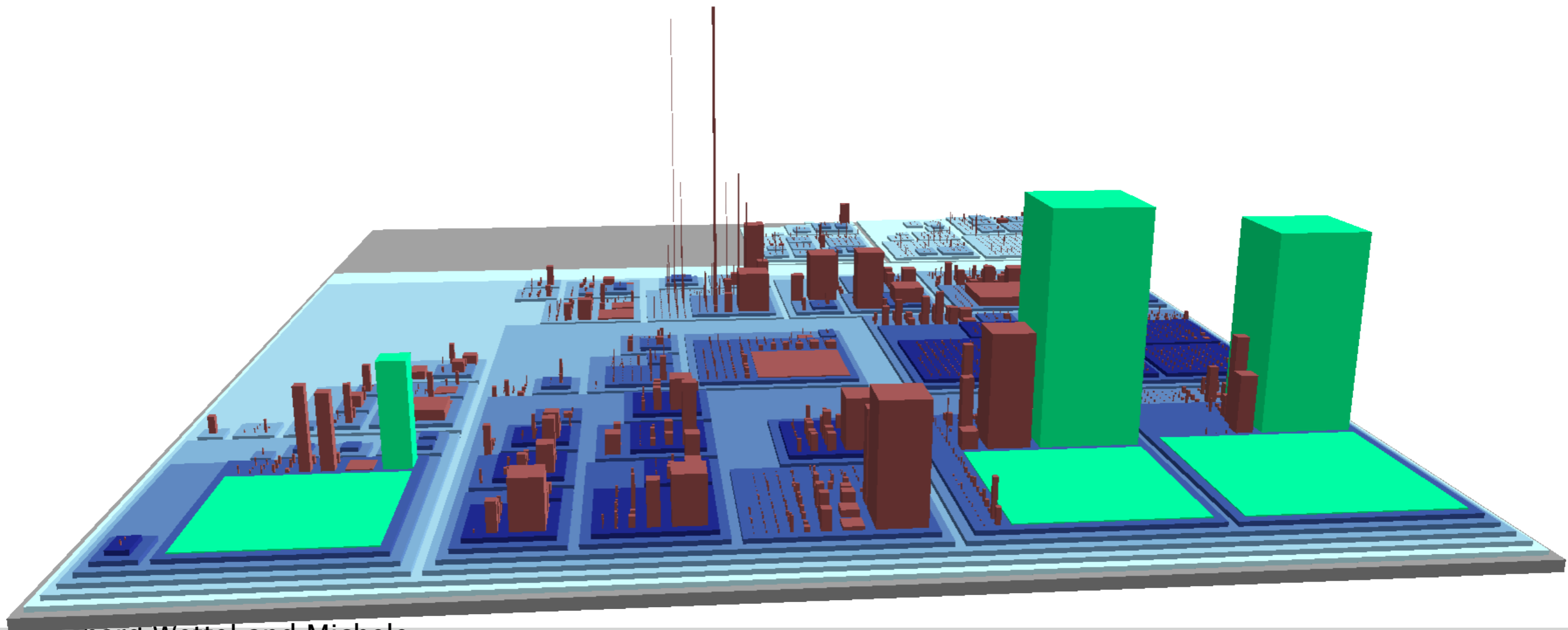
The twin towers



The twin towers

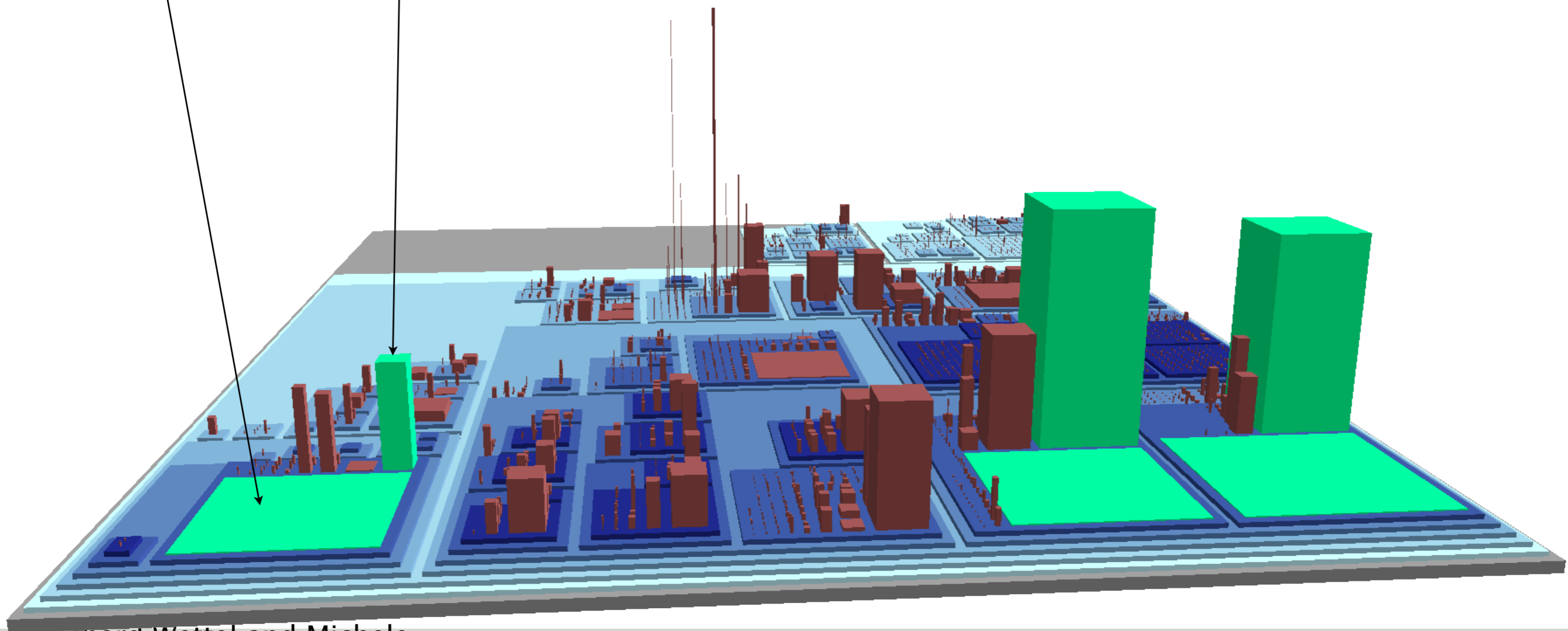


Hotels, parking lots, duplication



Hotels, parking lots, duplication

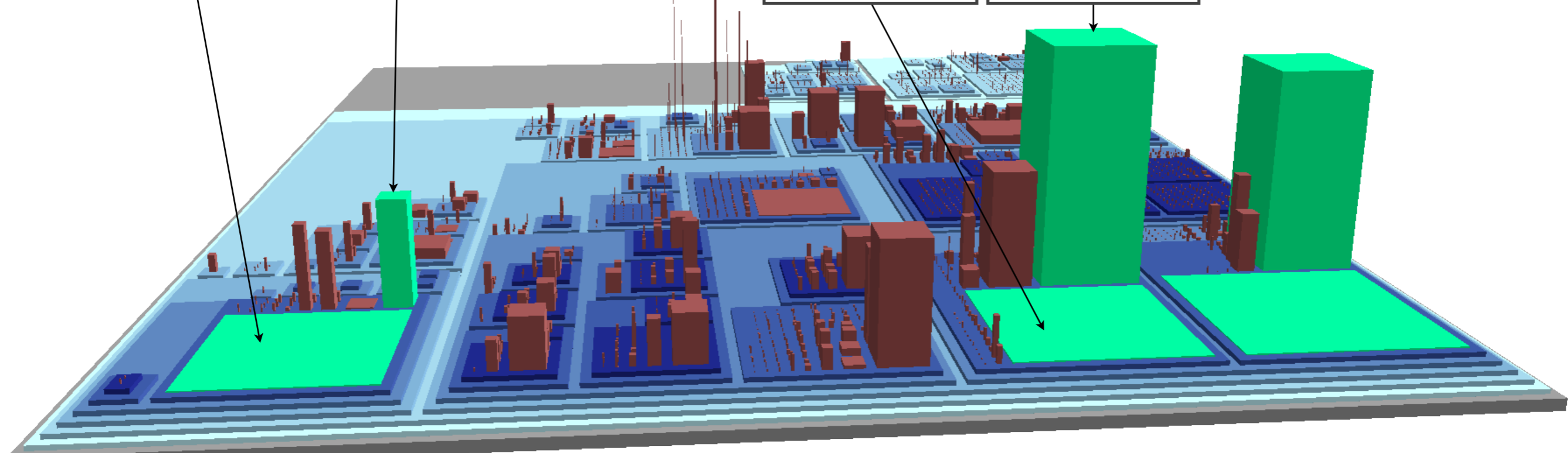
org.argouml.language. java.generator	
JavaTokenType	JavaRecognizer
146 attributes	24 attributes
0 methods	91 methods



Hotels, parking lots, duplication

org.argouml.language.java.generator	
JavaTokenType	JavaRecognizer
146 attributes	24 attributes
0 methods	91 methods

org.argouml.uml.reveng.class file	
STDCTokenTy	CPPParser
152 attributes	85 attributes
0 methods	204 methods

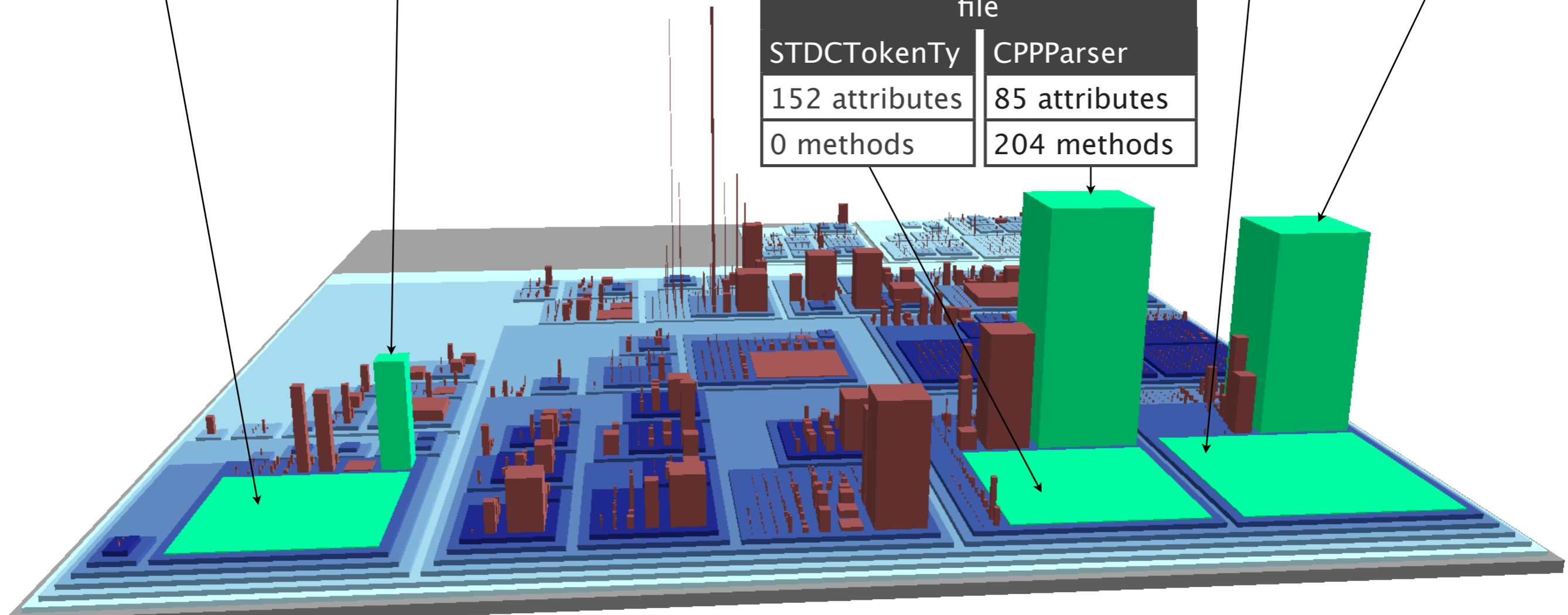


Hotels, parking lots, duplication

org.argouml.language.java.generator	
JavaTokenType	JavaRecognizer
146 attributes	24 attributes
0 methods	91 methods

org.argouml.uml.reveng.java	
JavaTokenTyp	JavaRecogniz
173	79 attributes
0 methods	176 methods

org.argouml.uml.reveng.class file	
STDCTokenTy	CPPParser
152 attributes	85 attributes
0 methods	204 methods



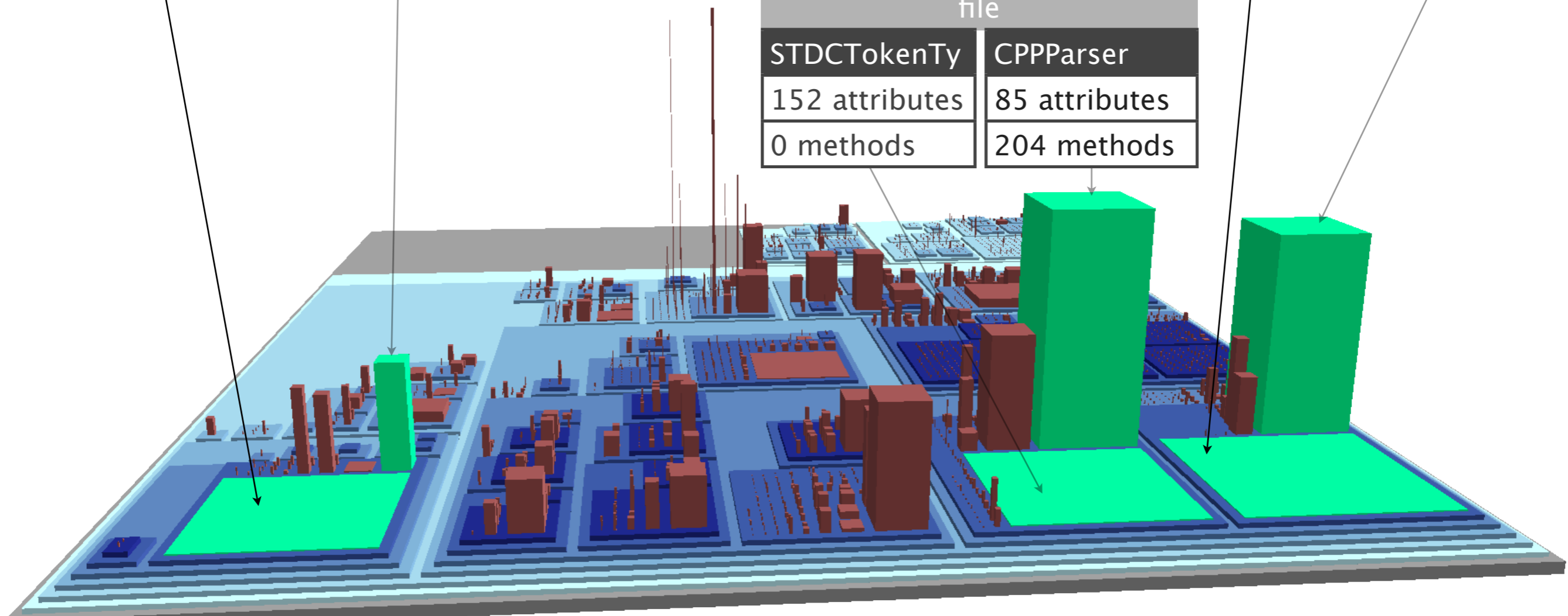
Hotels, parking lots, duplication

org.argouml.language.java.generator	
JavaTokenType	JavaRecognizer
146 attributes	24 attributes
0 methods	91 methods

JavaTokenType
145 attributes
0 methods

org.argouml.uml.reveng.java	
JavaTokenTyp	JavaRecogniz
173	79 attributes
0 methods	176 methods

org.argouml.uml.reveng.class file	
STDCTokenTy	CPPParser
152 attributes	85 attributes
0 methods	204 methods



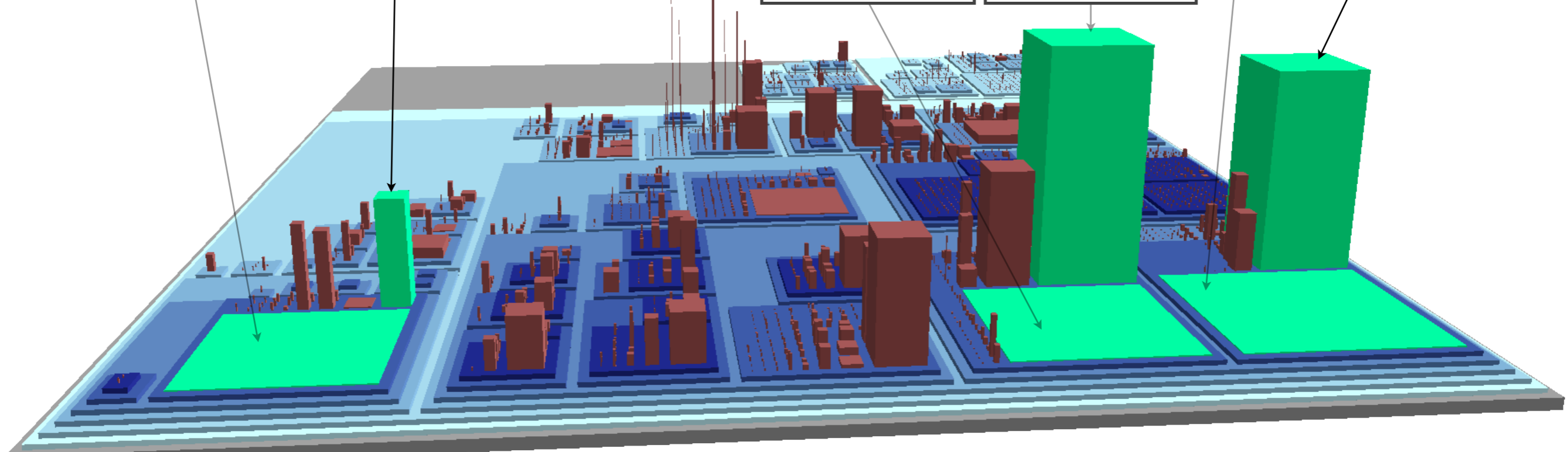
Hotels, parking lots, duplication

org.argouml.language.java.generator	
JavaTokenType	JavaRecognizer
146 attributes	24 attributes
0 methods	91 methods

JavaTokenType	JavaRecognizer
145 attributes	22 attributes
0 methods	88 methods

org.argouml.uml.reveng.java	
JavaTokenTyp	JavaRecogniz
173	79 attributes
0 methods	176 methods

org.argouml.uml.reveng.class file	
STDCTokenTy	CPPParser
152 attributes	85 attributes
0 methods	204 methods





Looking at the design

On software design...

On software design...

Definitions

bad smells [Beck & Fowler 1998], design flaws

heuristics [Riel 2000], principles [Martin 2002],
patterns [GoF 1995]

On software design...

Definitions

bad smells [Beck & Fowler 1998], design flaws

heuristics [Riel 2000], principles [Martin 2002],
patterns [GoF 1995]

Techniques

detection strategies for the design flaws

On software design...

Definitions

bad smells [Beck & Fowler 1998], design flaws

heuristics [Riel 2000], principles [Martin 2002],
patterns [GoF 1995]

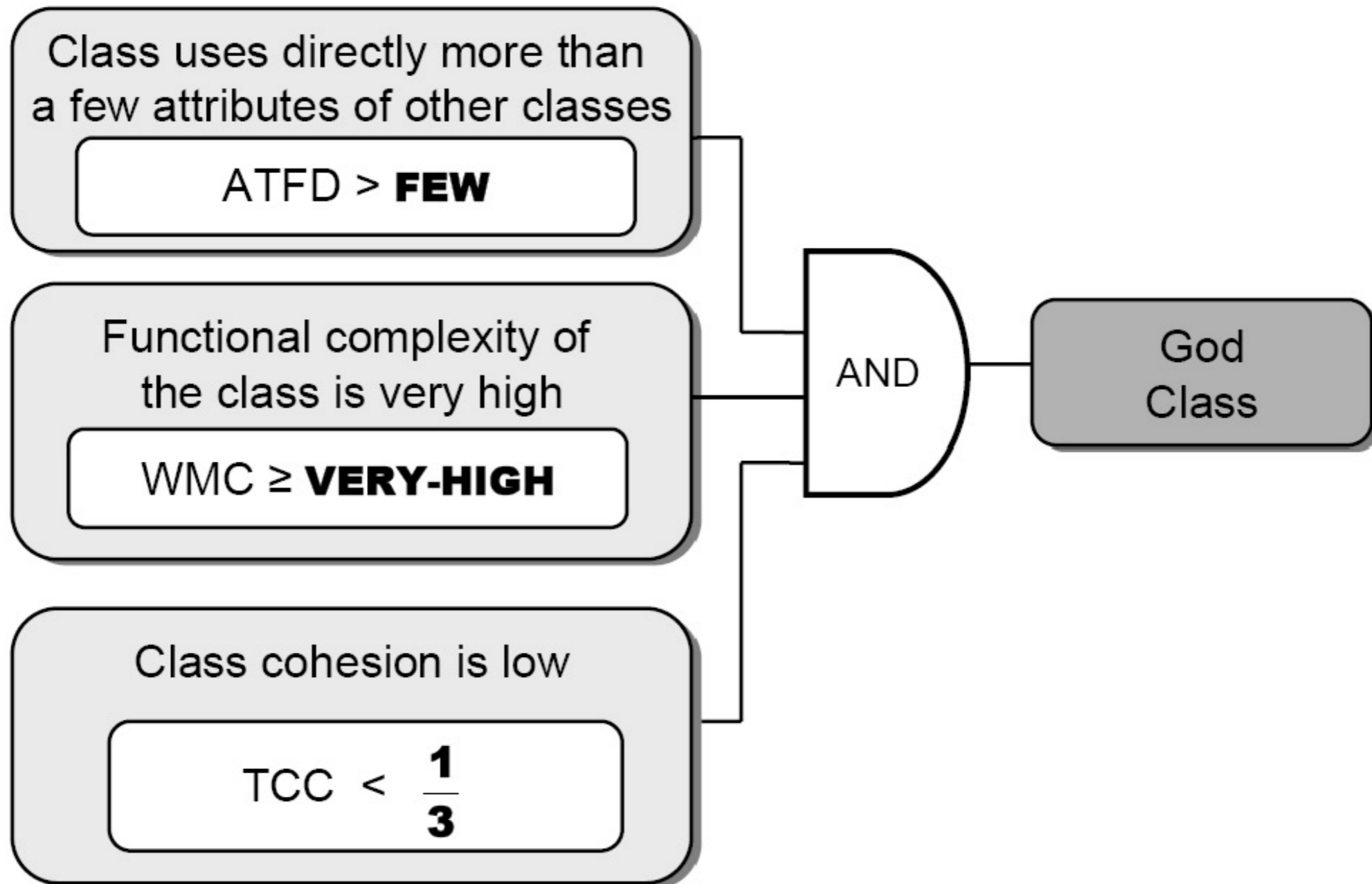
Techniques

detection strategies for the design flaws

Visualization

?

The God Class detection strategy



God classes of JDK 1.5 in Moose

JDK1.5 (Model)

- All namespaces (121 Namespaces)
- All classes (4715 Classes)
- All system classes (1966 Classes)
- All accesses (99180 Accesses)
- All packages (137 Packages)
- All invocations (39947 Invocations)
- All localvariables (16111 LocalVariables)
- All attributes (7716 Attributes)
- All inheritance definitions (3232 InheritanceDefinitions)
- All methods (19379 Methods)
- All functions (1 Function)
- All formalparameters (20464 FormalParameters)

Property	Value	Description
Entities	21100	Entities
NOCs	1966	Number of model (non stub) classes
NOM	18355	Number of model (non stub) methods
numberOfIt	21100	numberOfItems

All classes (4715 Classes)

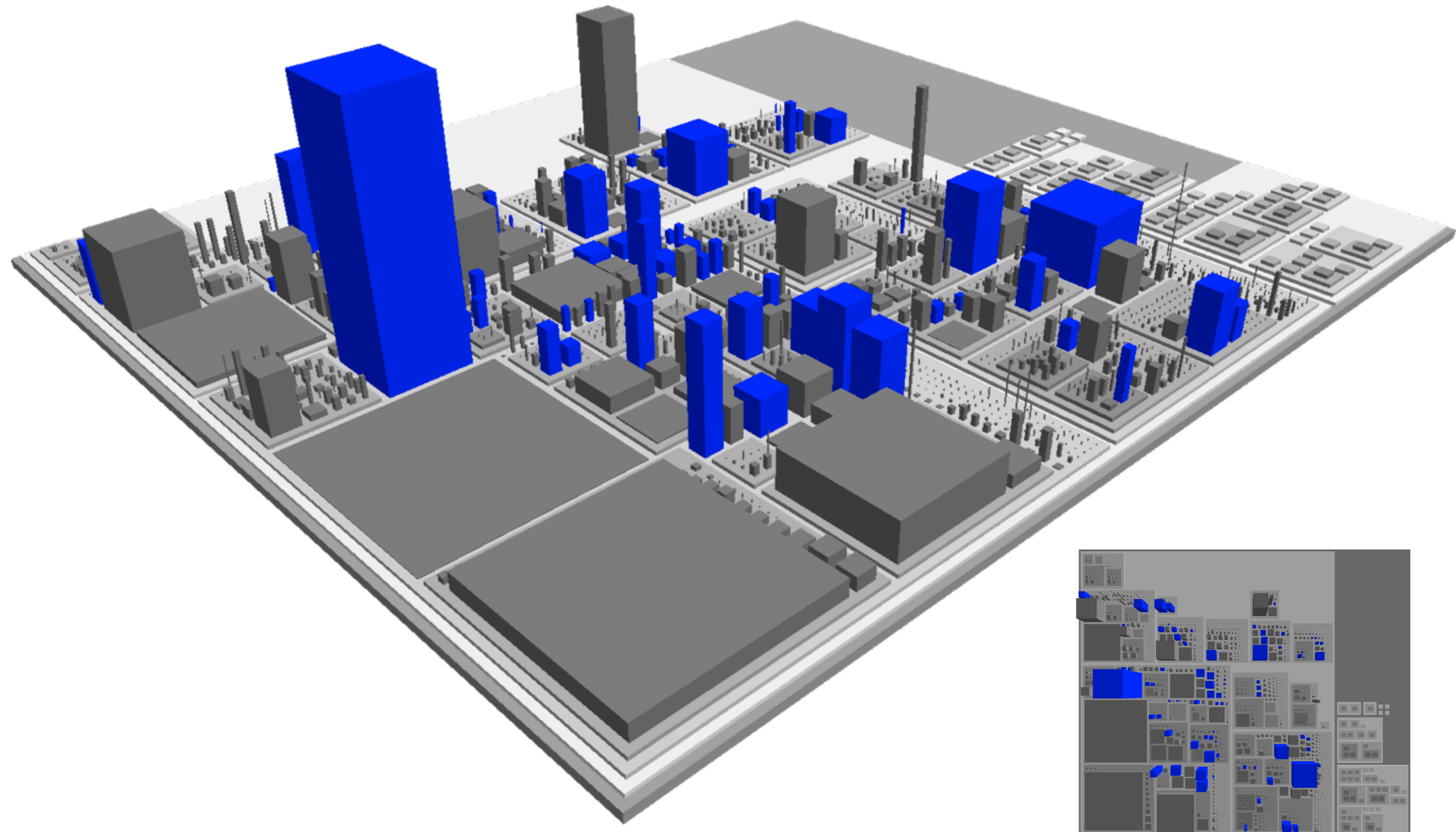
- java:awt:Applet\$AccessibleApplet
- java:awt:image:ColorModel
- java:awt:image:BufferStrategy
- java:awt:image:WritableRaster
- java:awt:image:Raster
- java:awt:image:ImageObserver
- java:awt:image:ImageProducer
- java:awt:image:VolatileImage
- java:awt:image:DirectColorModel
- java:awt:image:BufferedImage
- java:awt:image:BufferedImageOp
- java:awt:image:RenderedImage
- java:awt:image:AffineTransformOp
- java:awt:image:RasterOp
- java:awt:image:ColorConvertOp
- java:awt:image:AreaAveragingScaleFilter
- java:awt:image:ReplicateScaleFilter
- java:awt:image:BandCombineOp
- java:awt:image:BandedSampleModel
- java:awt:image:ComponentSampleModel
- java:awt:image:SampleModel
- java:awt:image:DataBuffer
- java:awt:image:WritableRenderedImage
- java:awt:image:IndexColorModel
- java:awt:image:ComponentColorModel
- java:awt:image:PixelInterleavedSampleModel
- java:awt:image:TileObserver
- java:awt:image:BufferedImageFilter
- java:awt:image:ImageFilter
- java:awt:image:DataBufferByte
- java:awt:image:DataBufferInt
- java:awt:image:ByteLookupTable
- java:awt:image:LookupTable
- java:awt:image:ConvolveOp

Group (81 Classes)

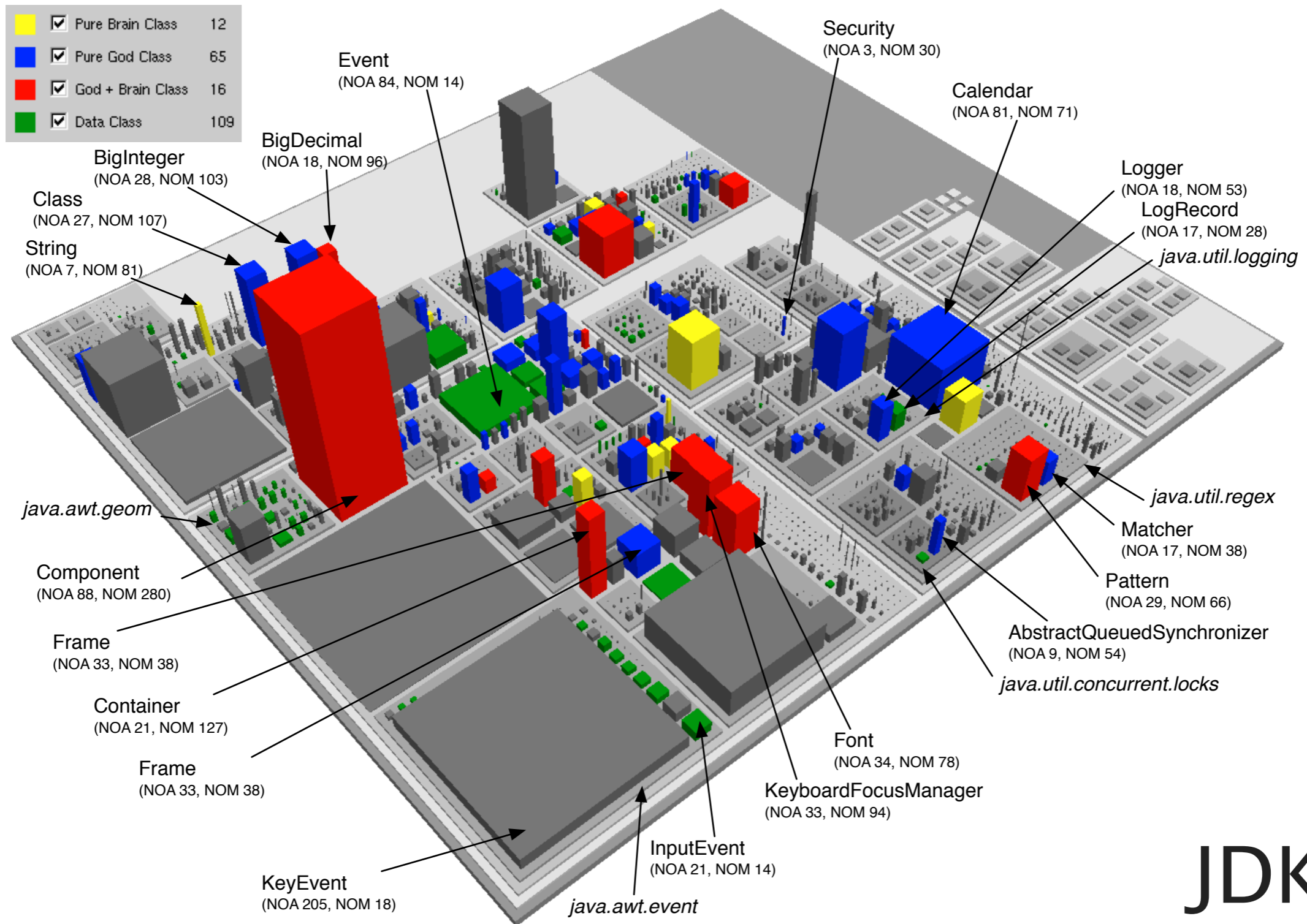
- java:lang:Class
- java:lang:ClassLoader
- java:lang:ConditionalSpecialCasing
- java:awt:image:ColorModel
- java:awt:image:Raster
- java:awt:image:BufferedImage
- java:awt:image:AffineTransformOp
- java:awt:image:ColorConvertOp
- java:awt:image:IndexColorModel
- java:awt:image:LookupOp
- java:awt:image:RescaleOp
- java:net:URL
- java:net:DatagramSocket
- java:net:MulticastSocket
- java:net:PlainSocketImpl
- java:net:ServerSocket
- java:net:Socket
- java:net:URI\$Parser
- java:net:URLStreamHandler
- java:util:ResourceBundle
- java:util:WeakHashMap
- java:util:HashMap
- java:util:TreeMap
- java:util:Calendar
- java:util:Scanner
- java:awt:AWTEvent
- java:awt:Component
- java:awt:AWTKeyStroke
- java:awt:BorderLayout
- java:awt:Container
- java:awt:CardLayout
- java:awt:MenuItem
- java:awt:Font
- java:awt:Toolkit

"God Class - Java" ATFD > 2 & WMC > 47 & TCC < 0.33

God classes of JDK in CodeCity

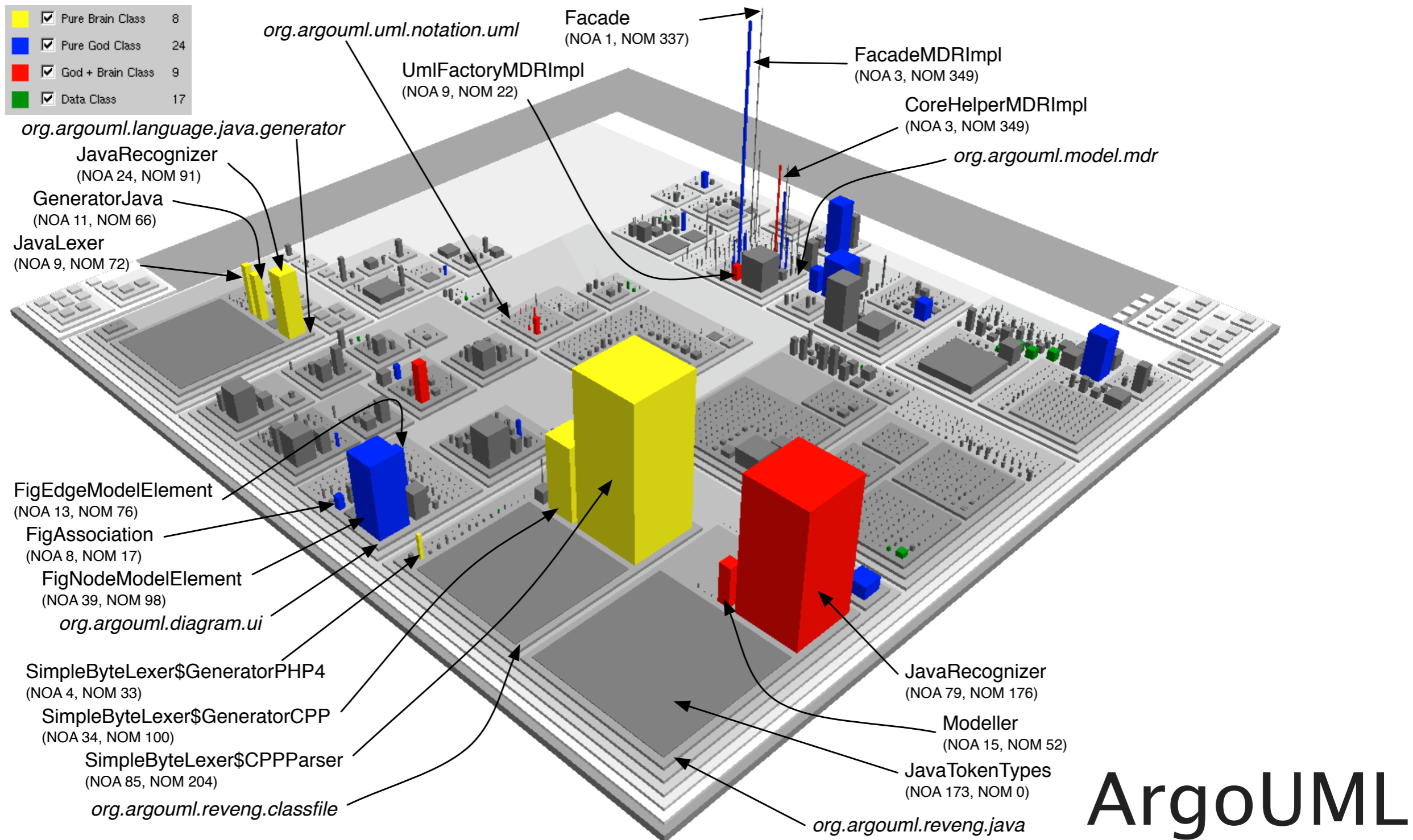


Class-level disharmony maps

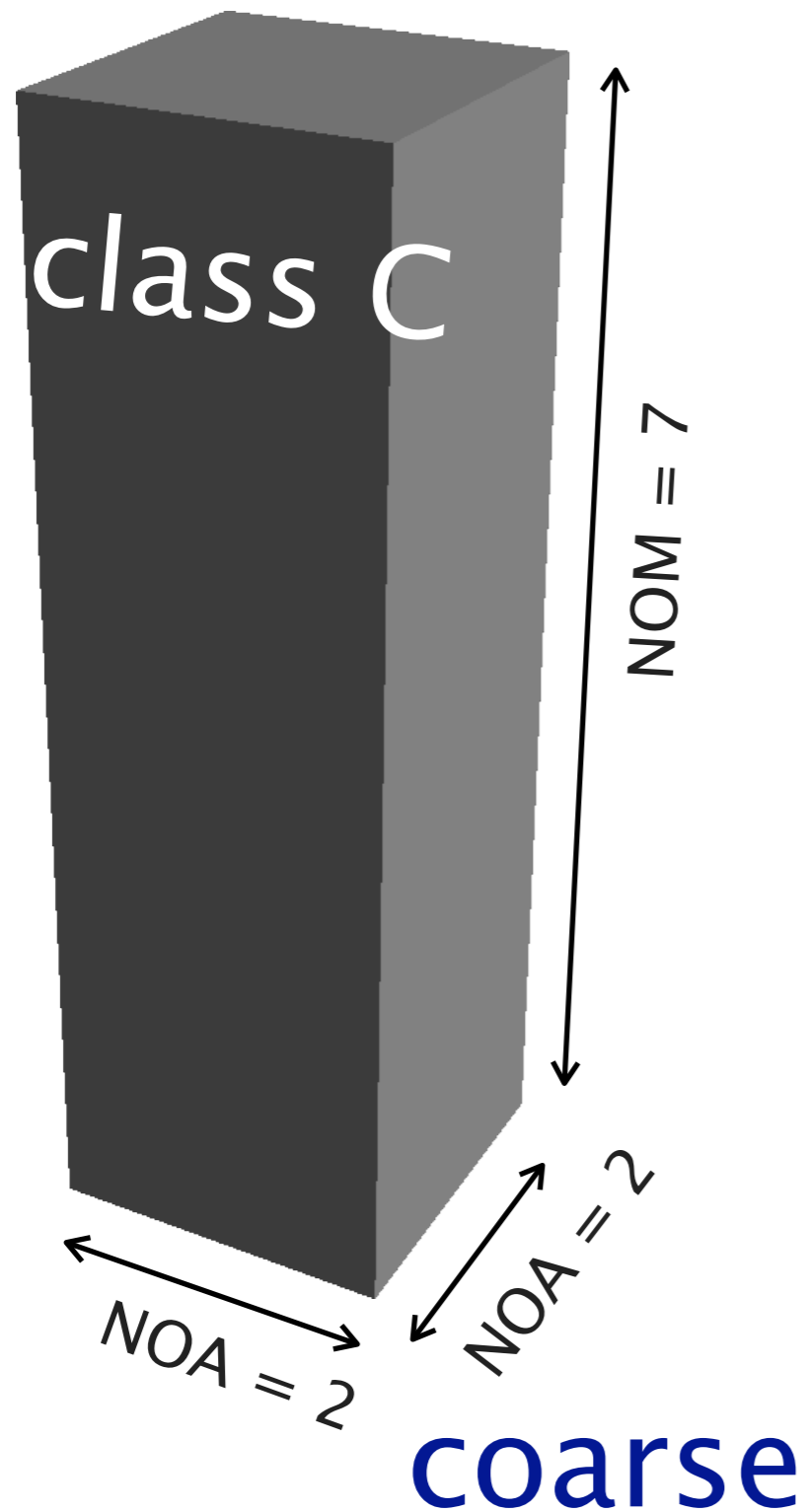


JDK 1.5

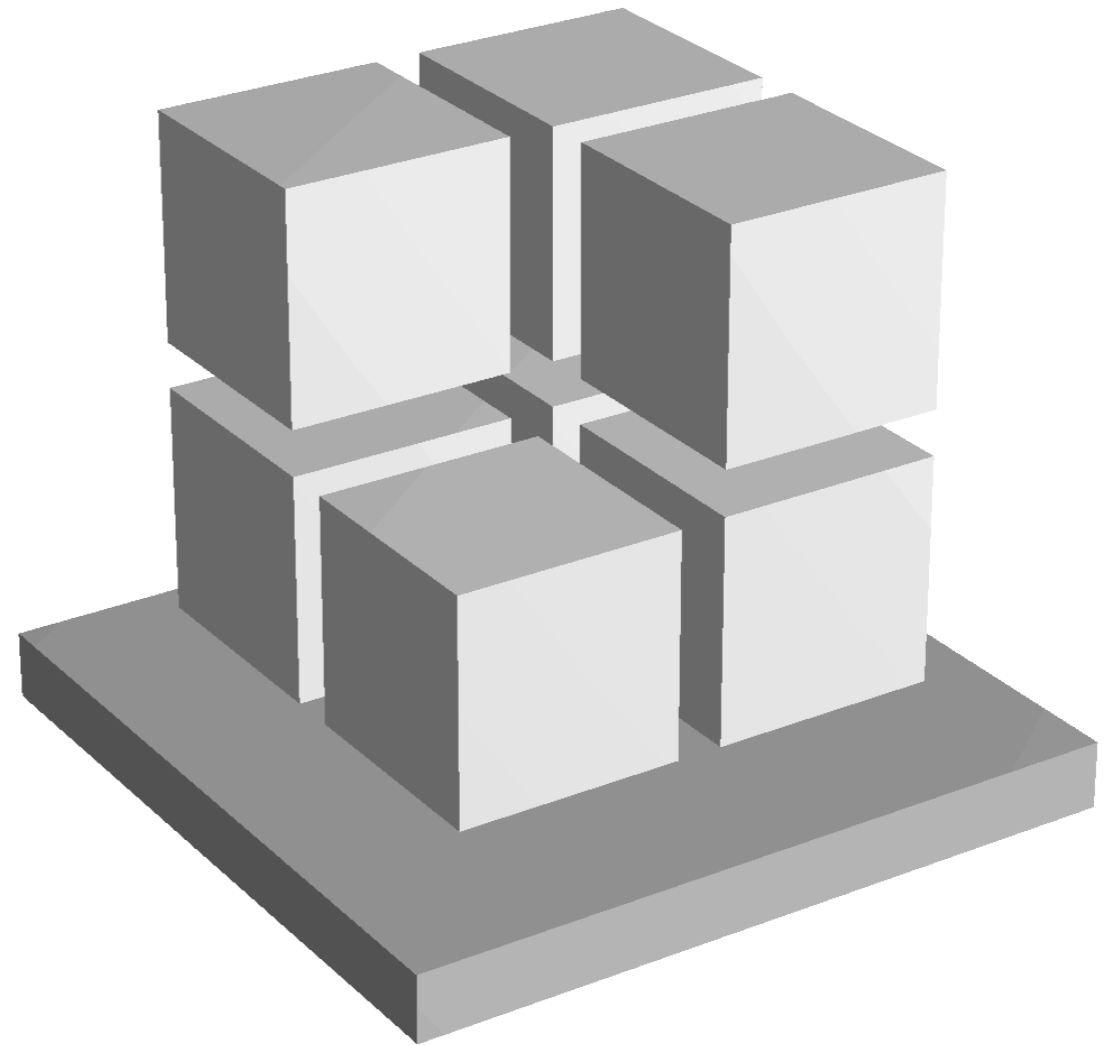
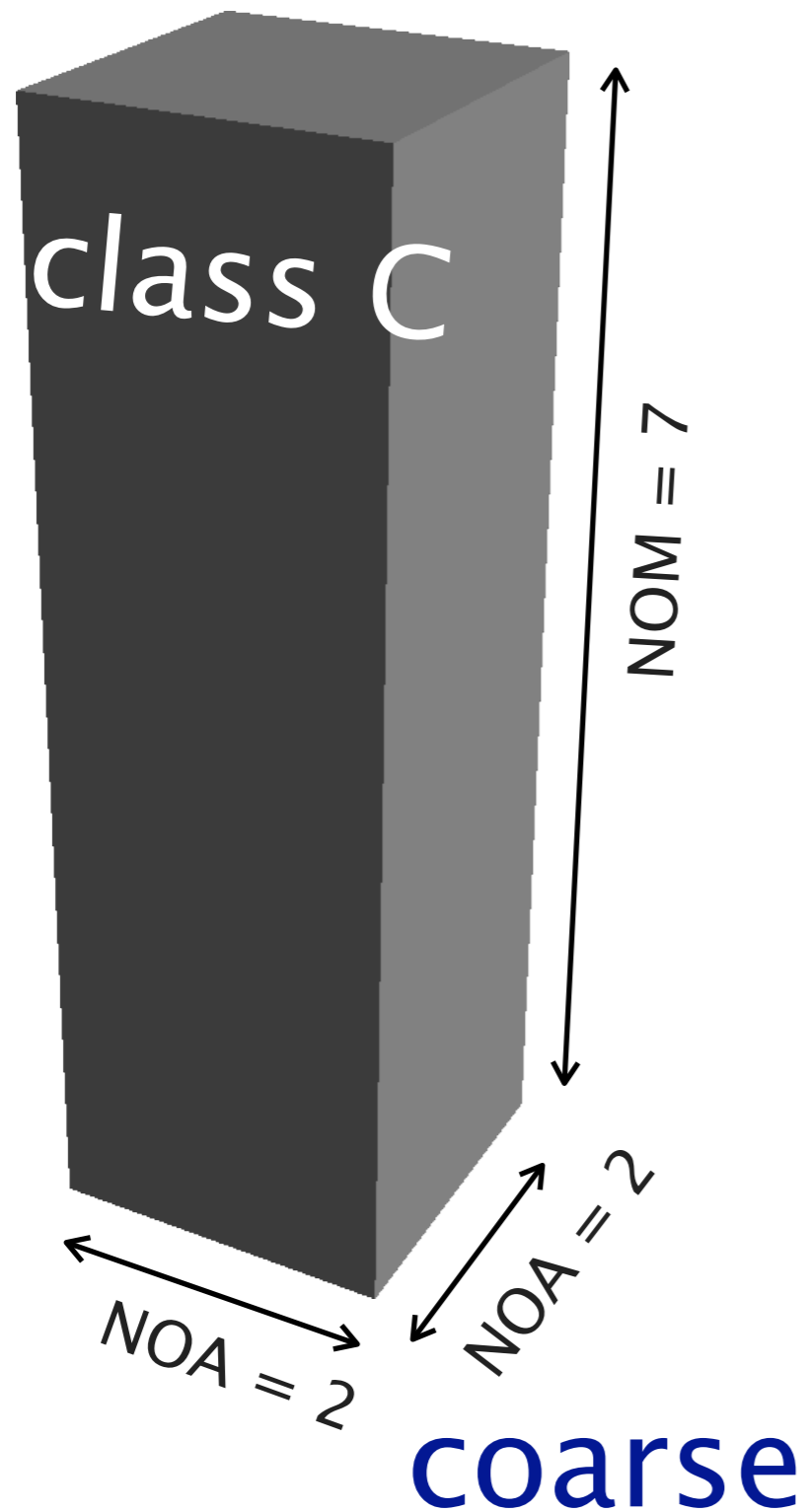
Class-level disharmony ma



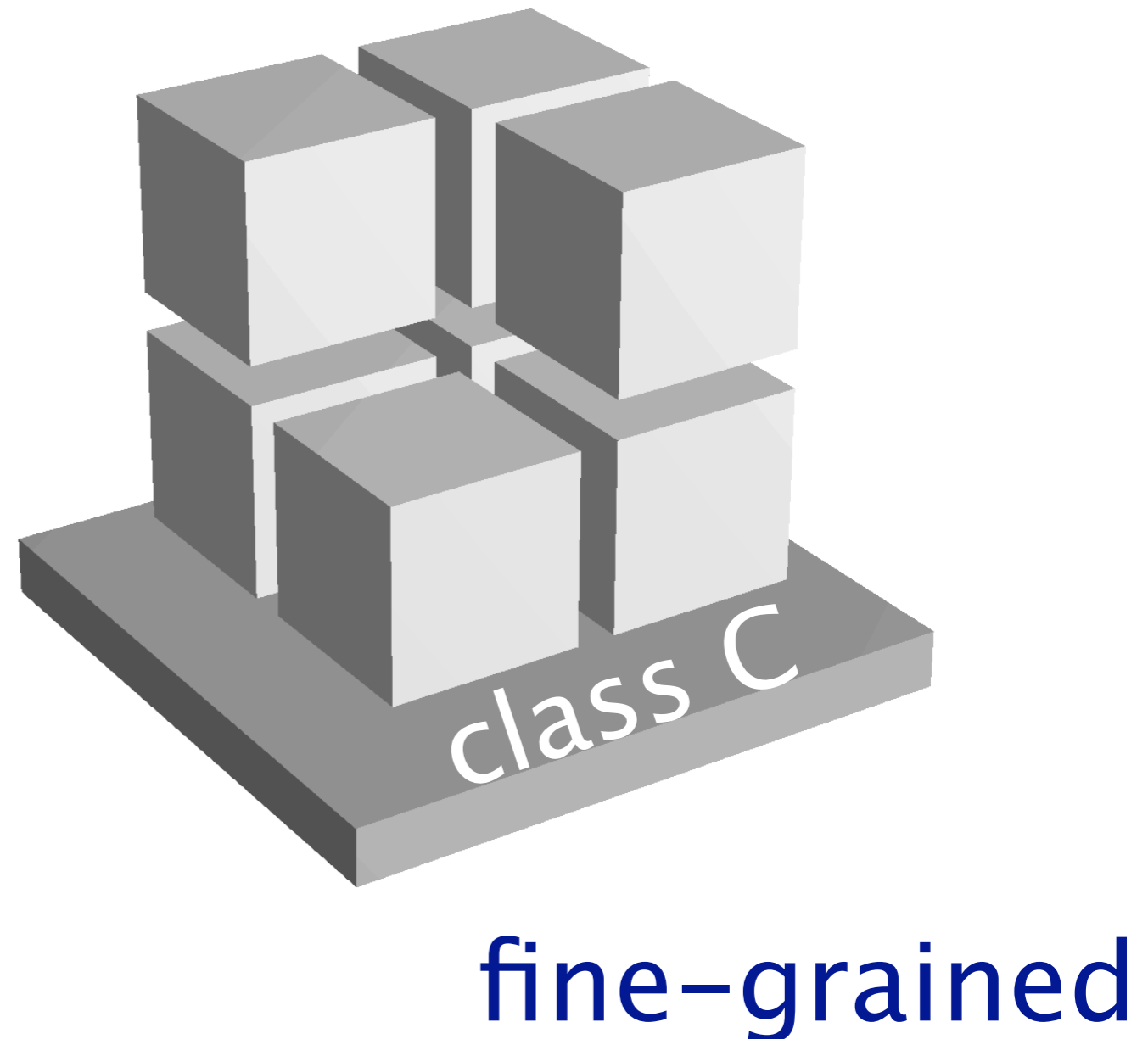
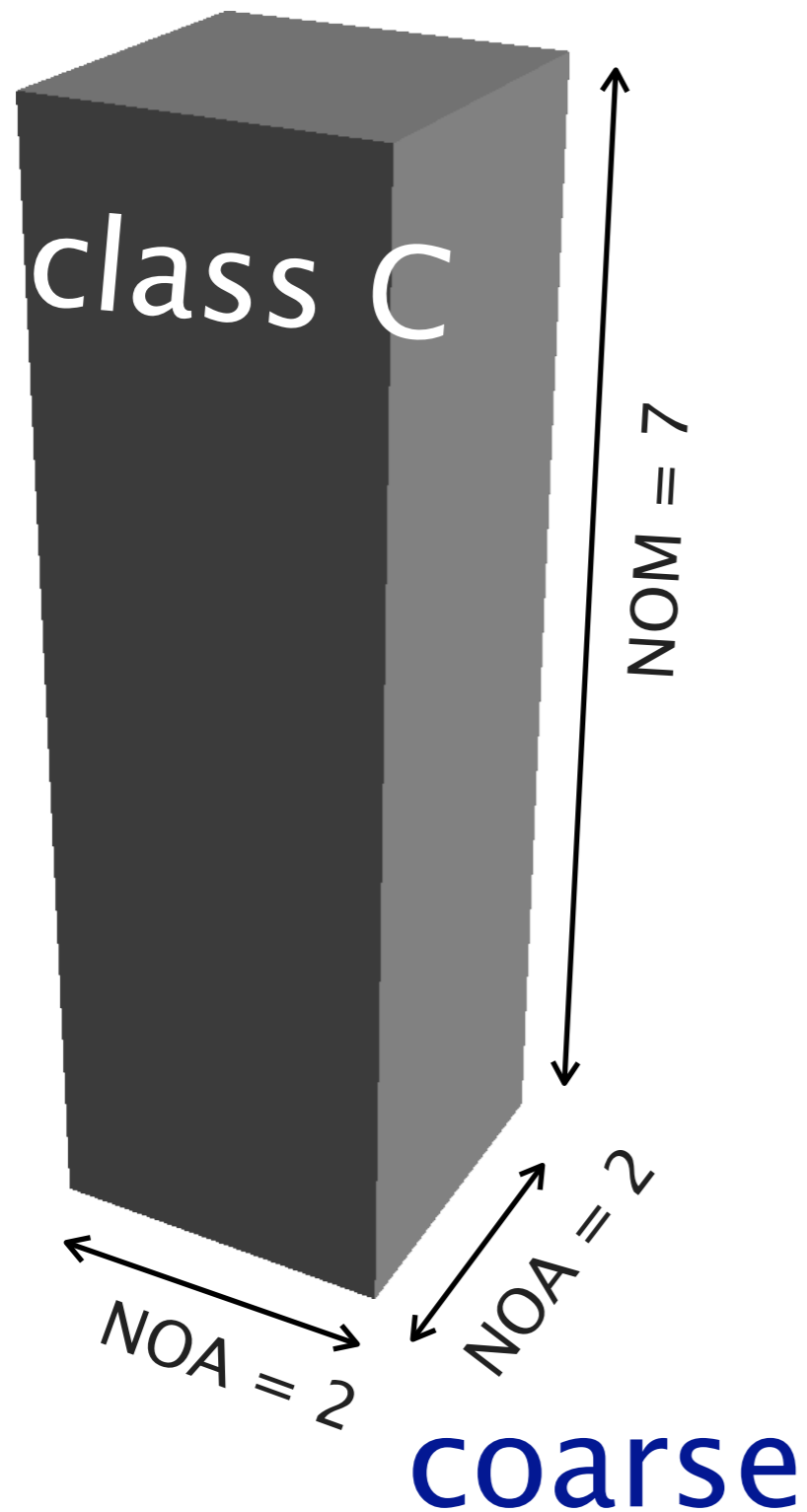
Granularity of representation



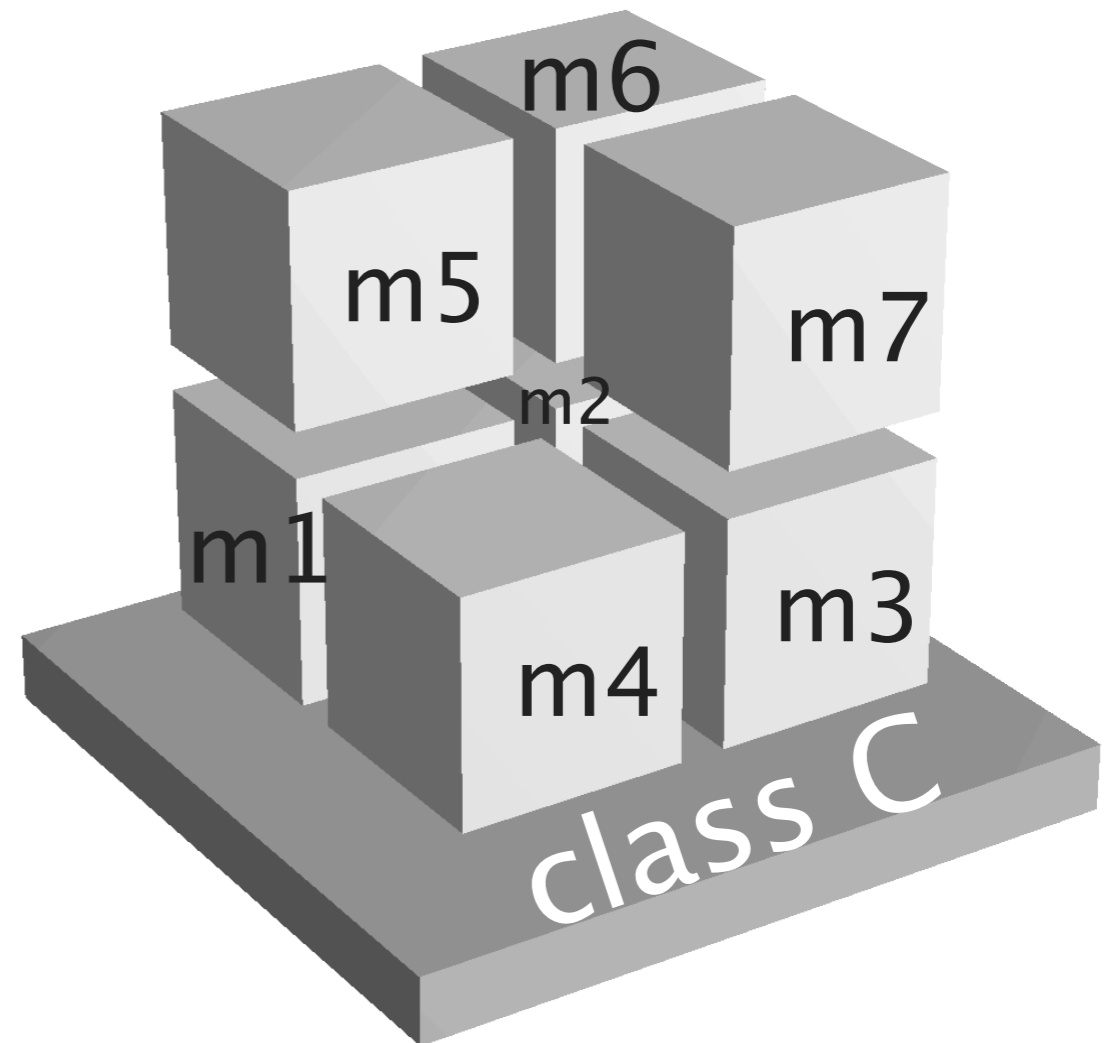
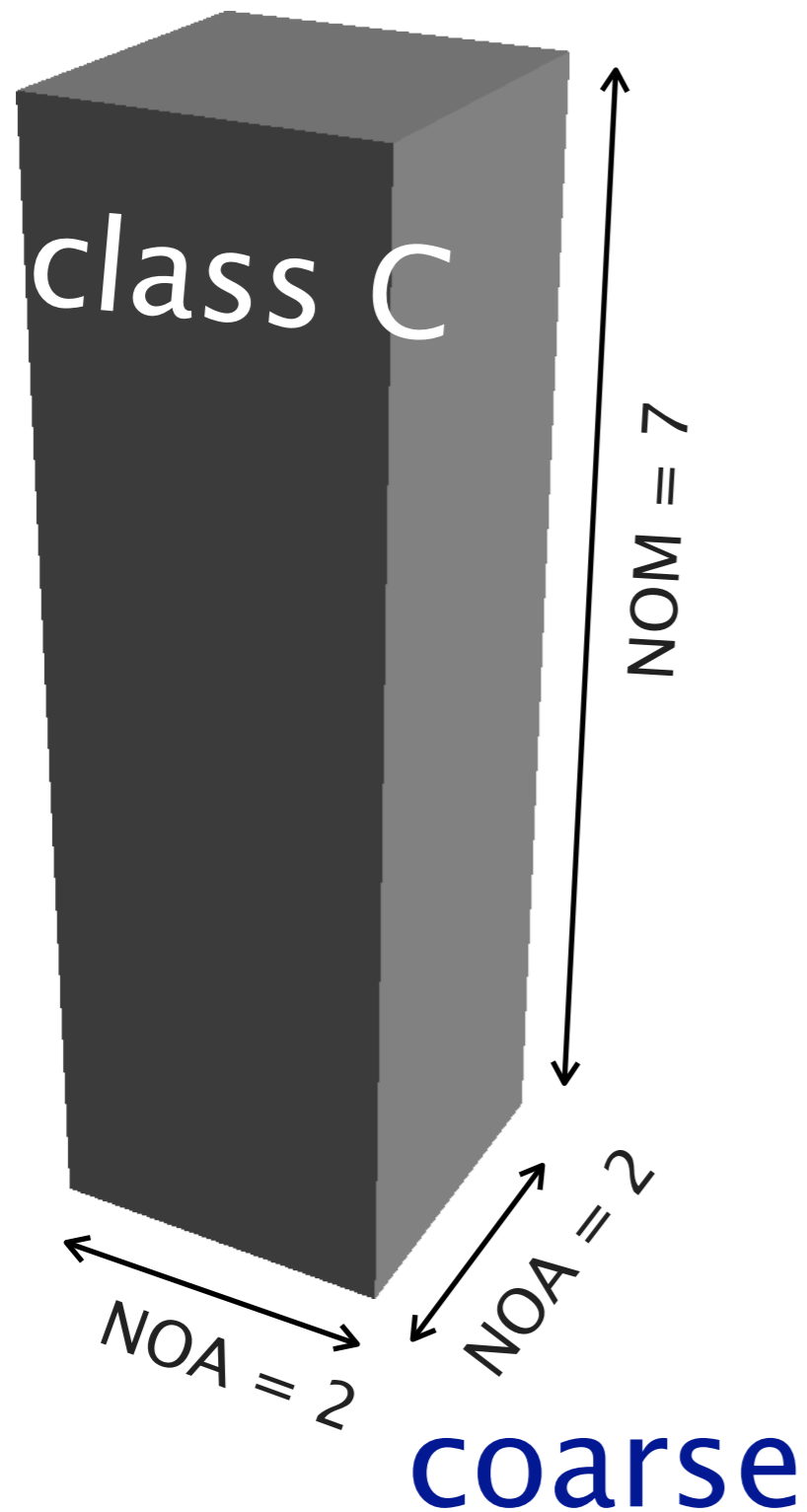
Granularity of representation



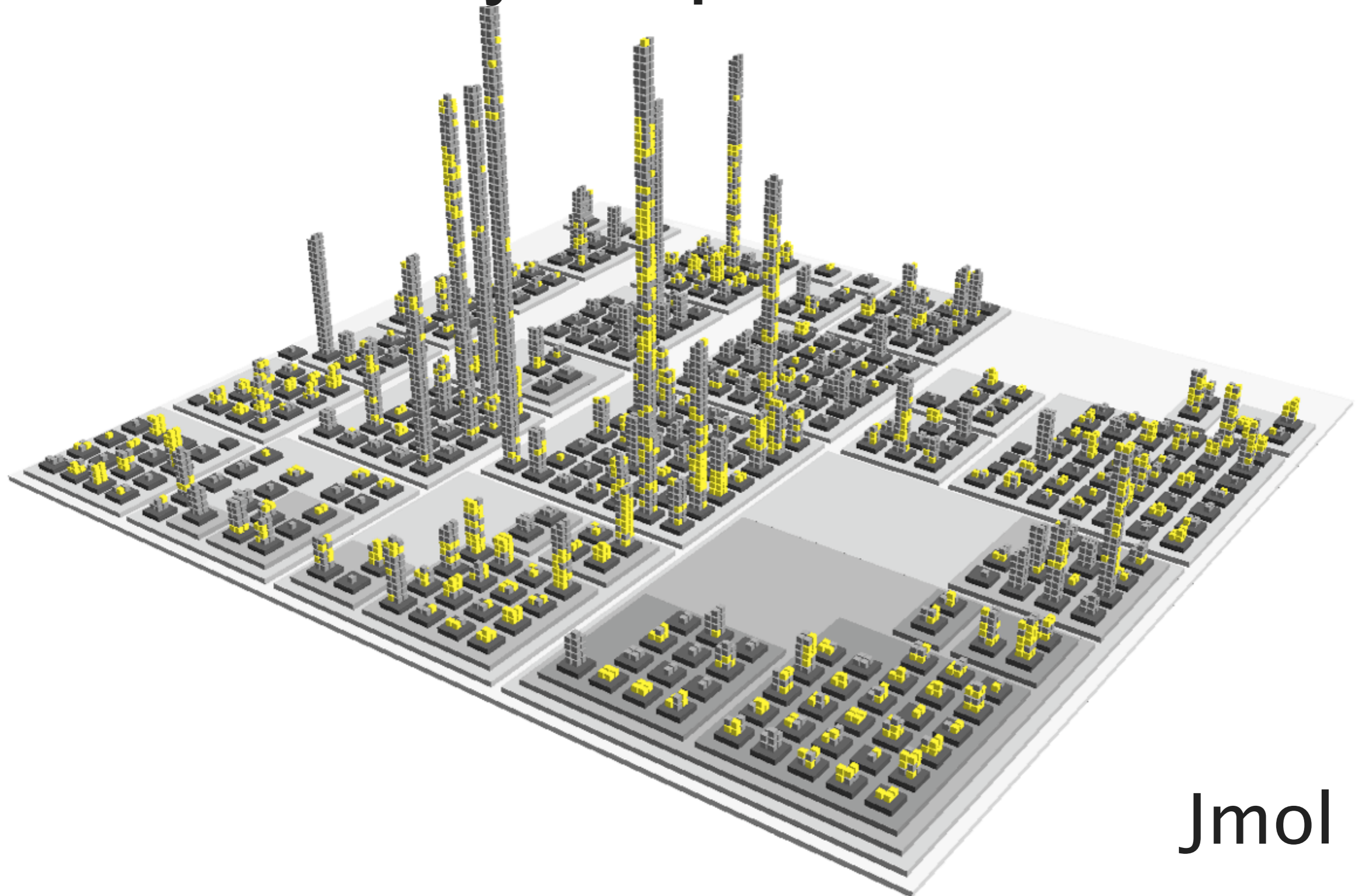
Granularity of representation



Granularity of representation

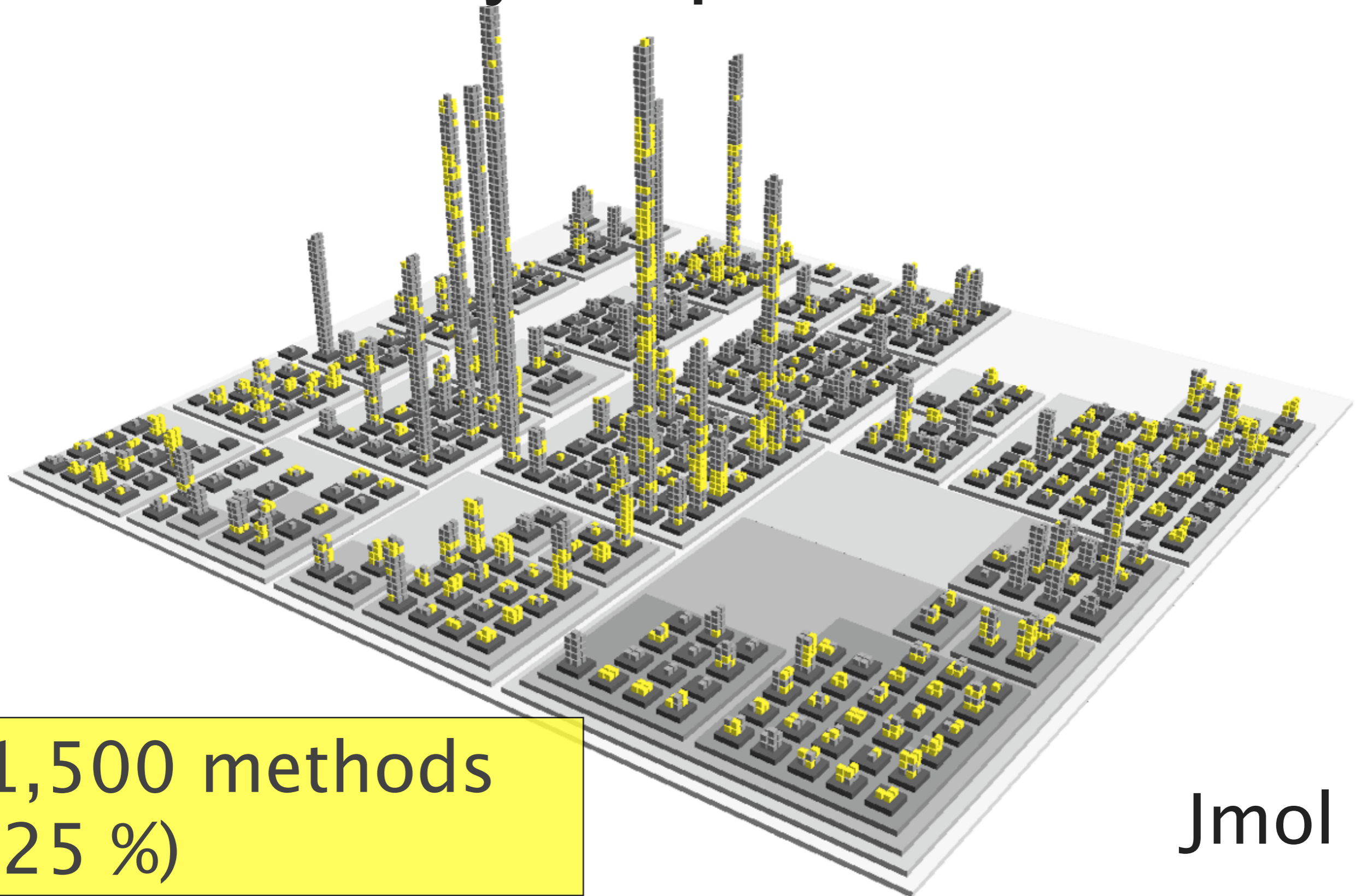


Feature envy map



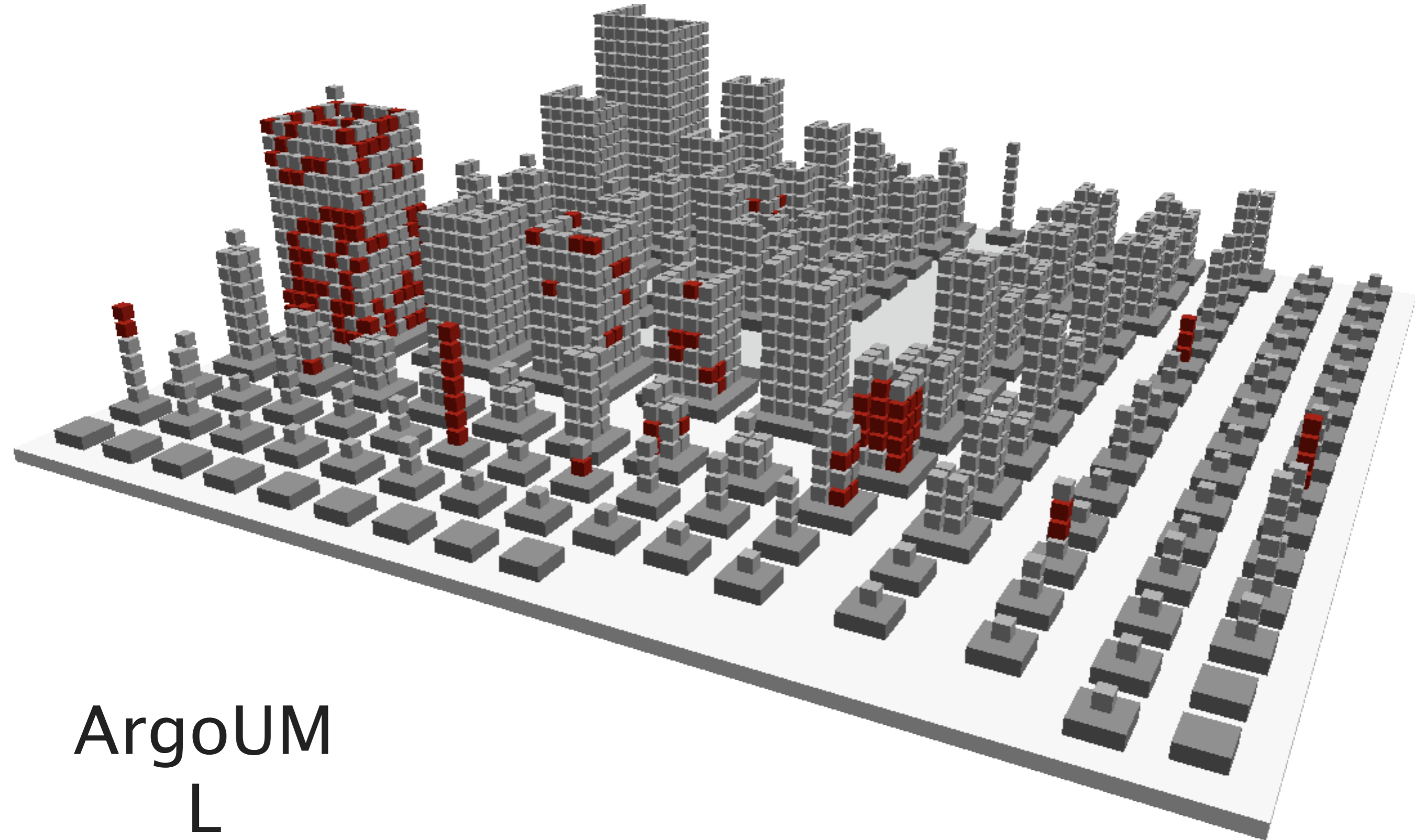
Jmol

Feature envy map

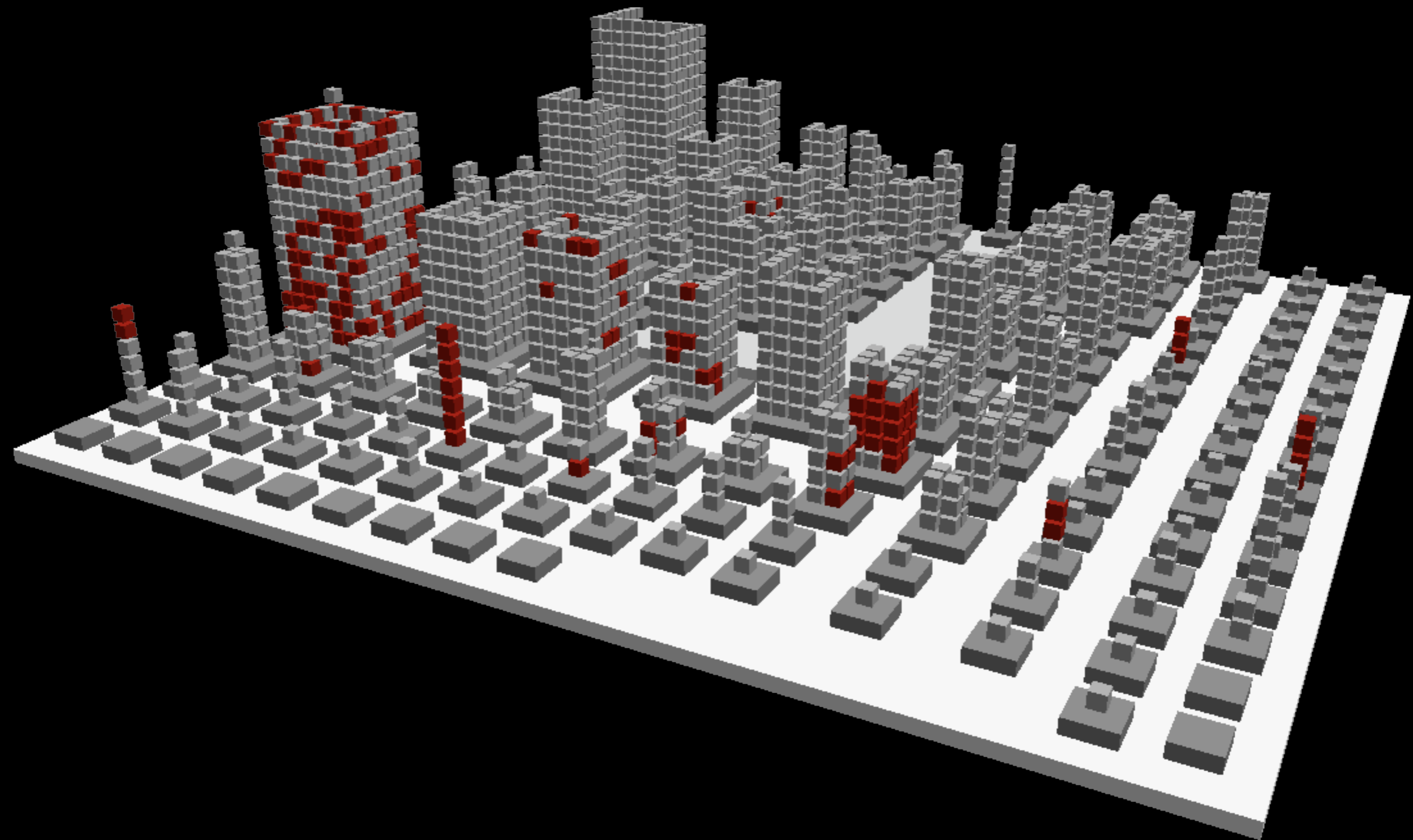


Jmol

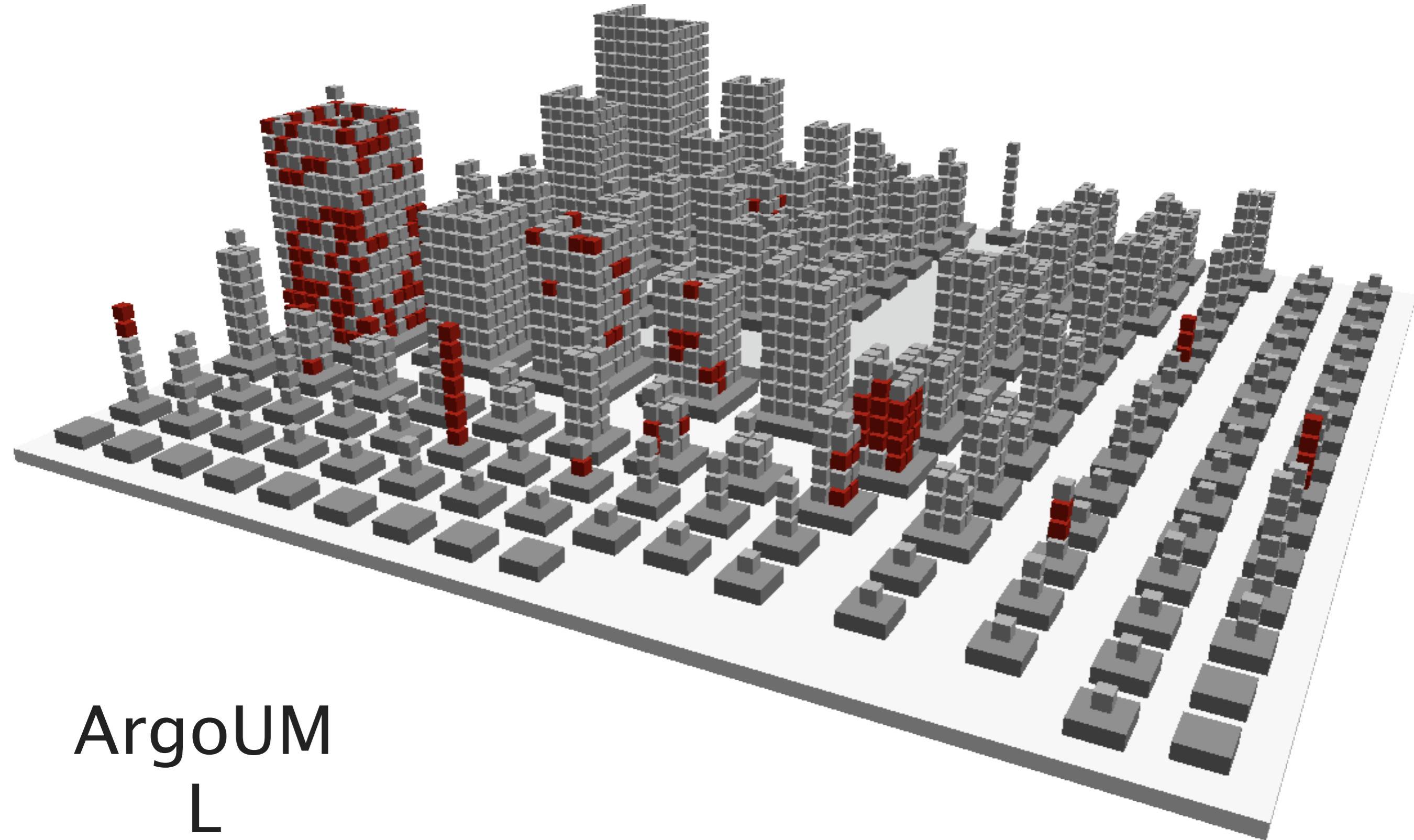
Shotgun surgery map



ArgoUM
L



Shotgun surgery map



ArgoUM
L



Exploring the history

Modeling history: Hismo

Class

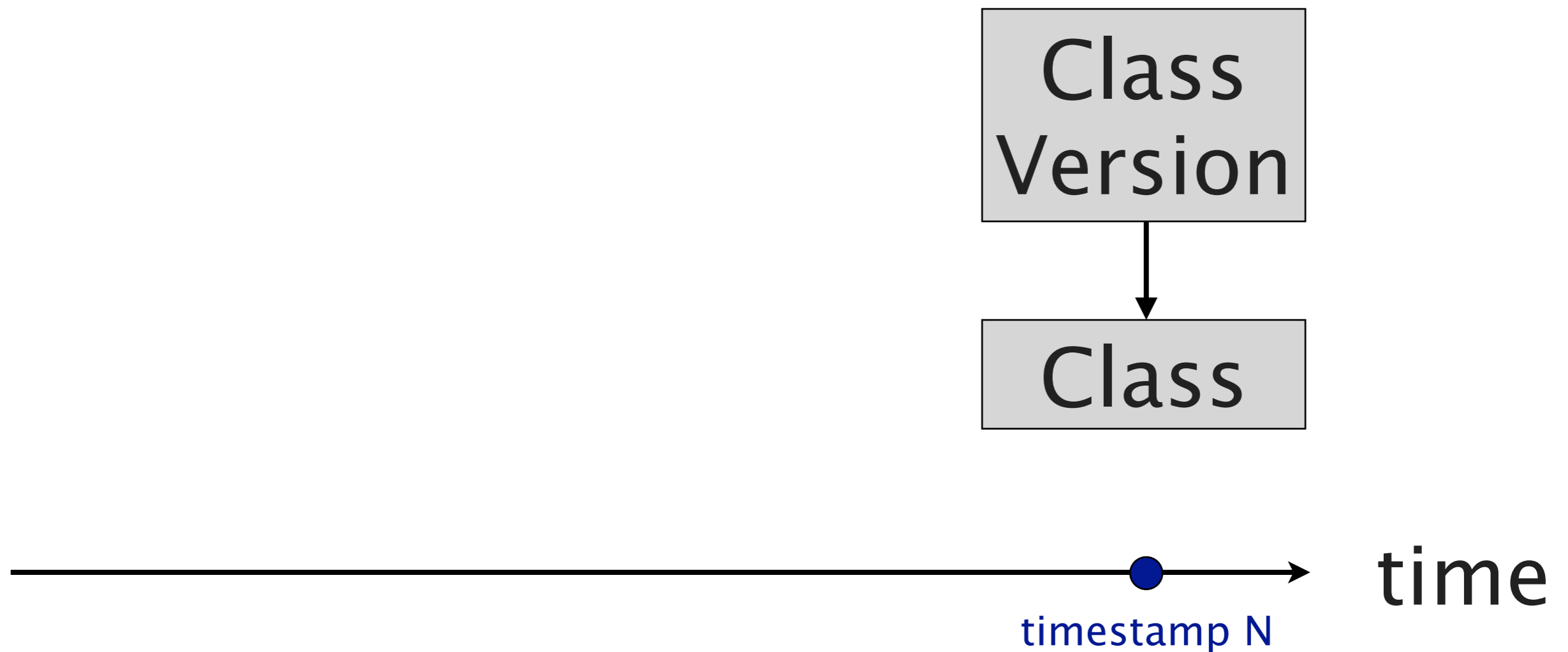
Modeling history: Hismo



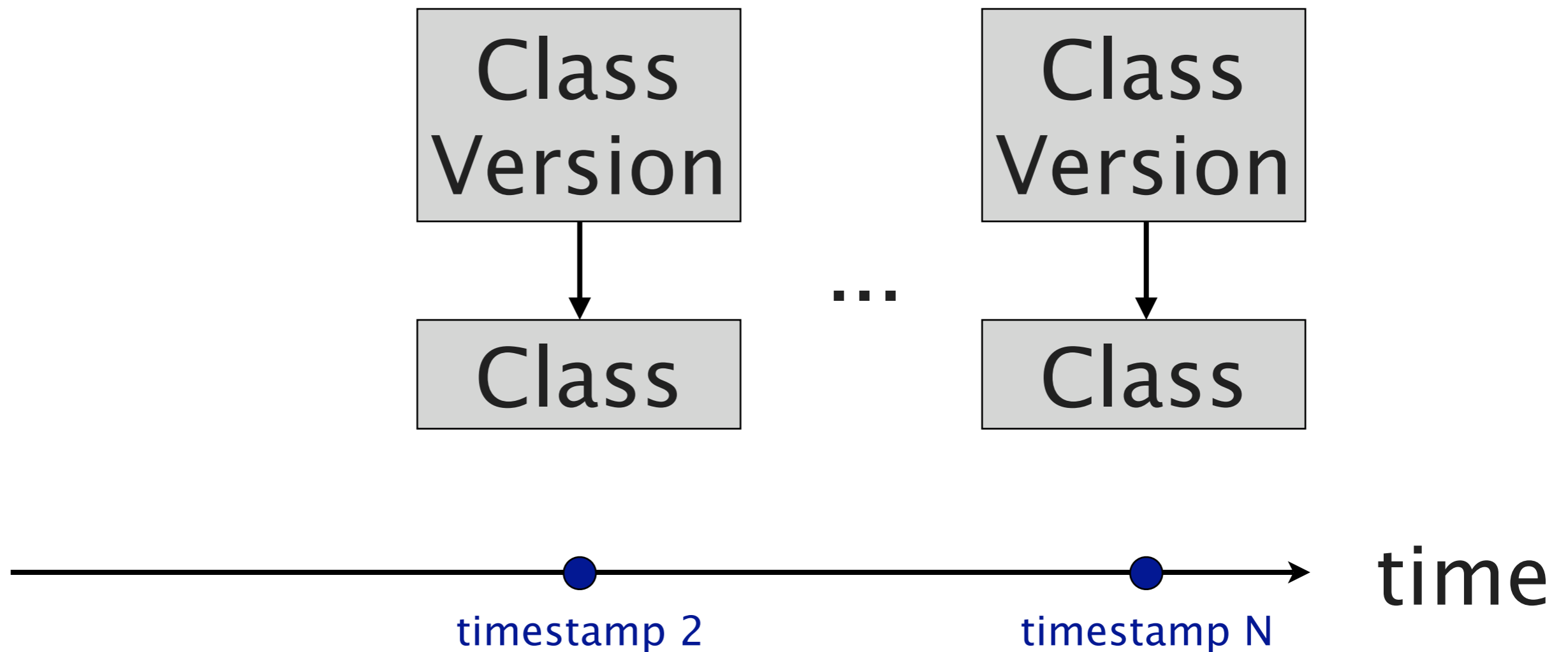
Class

time

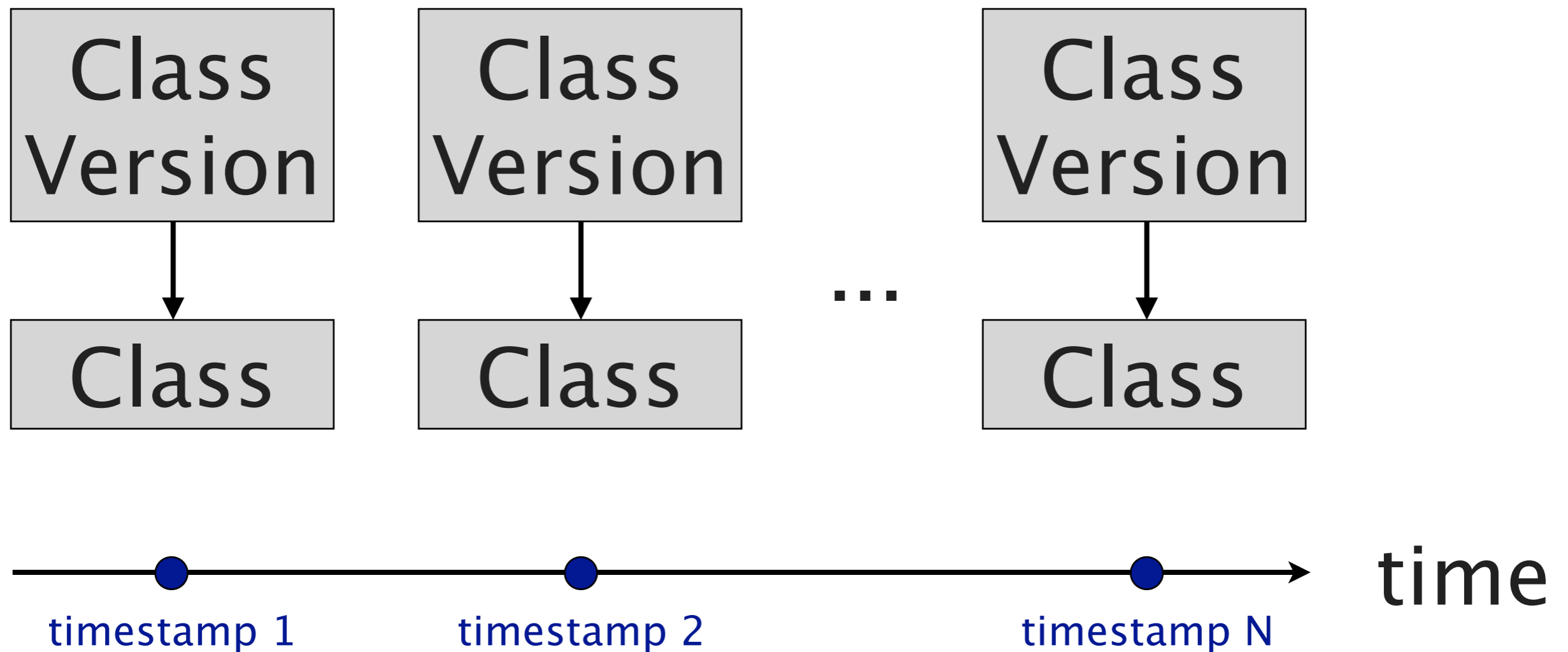
Modeling history: Hismo



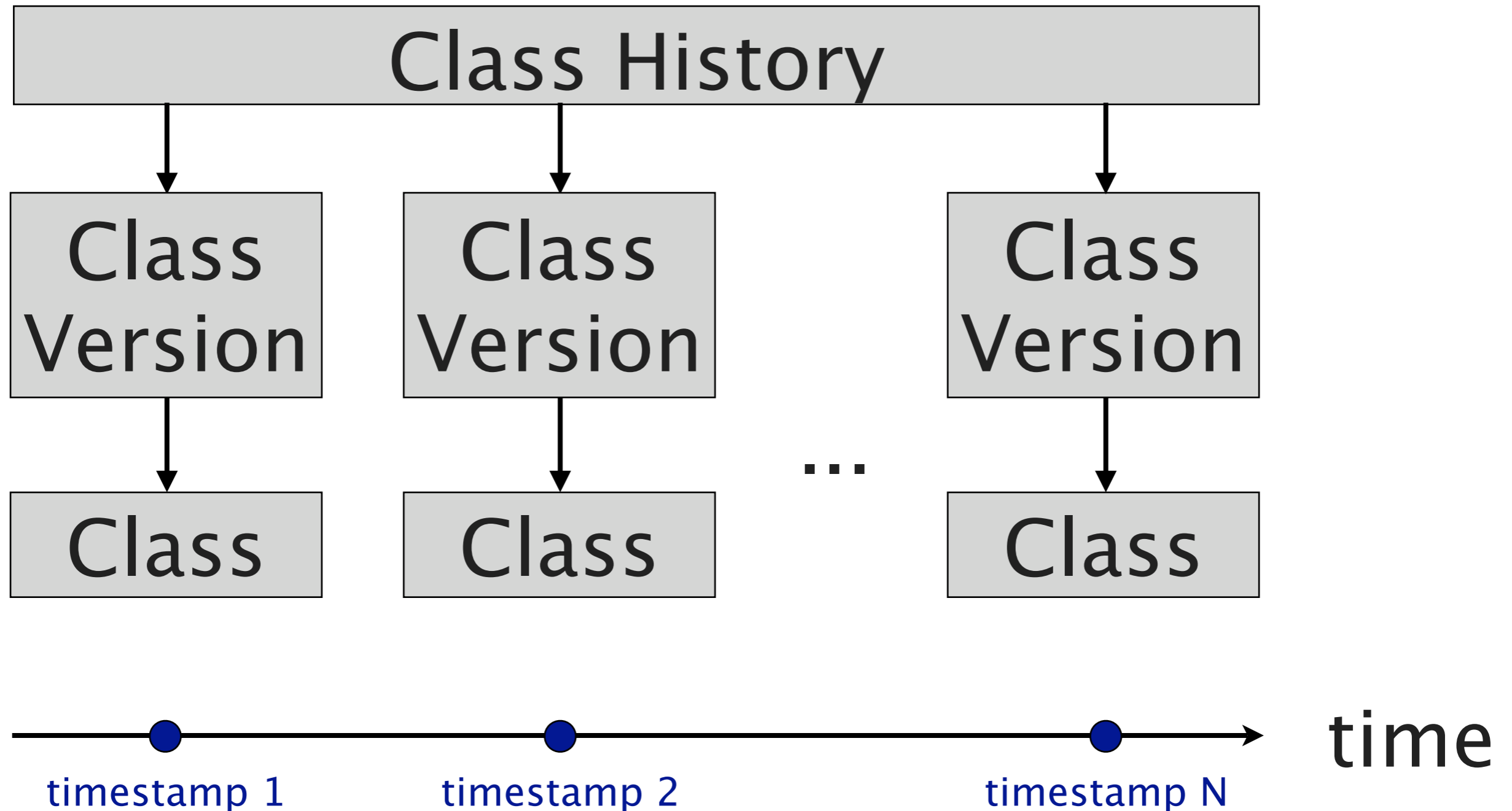
Modeling history: Hismo



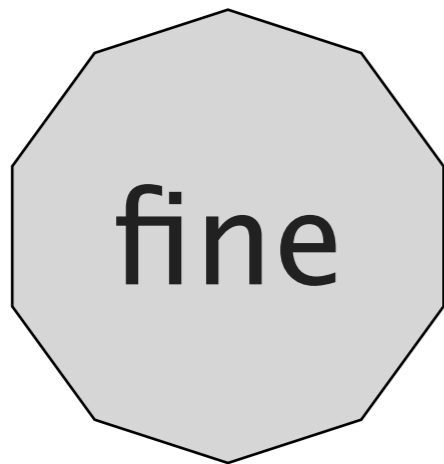
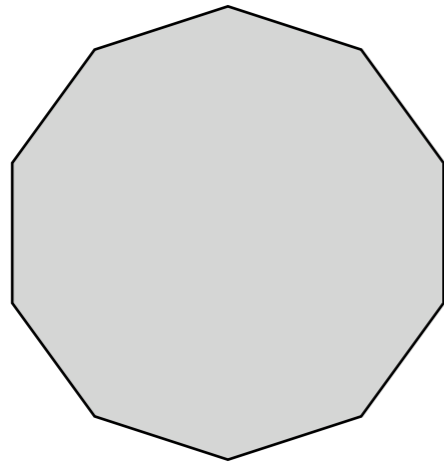
Modeling history: Hismo



Modeling history: Hismo



Visualizing evolution

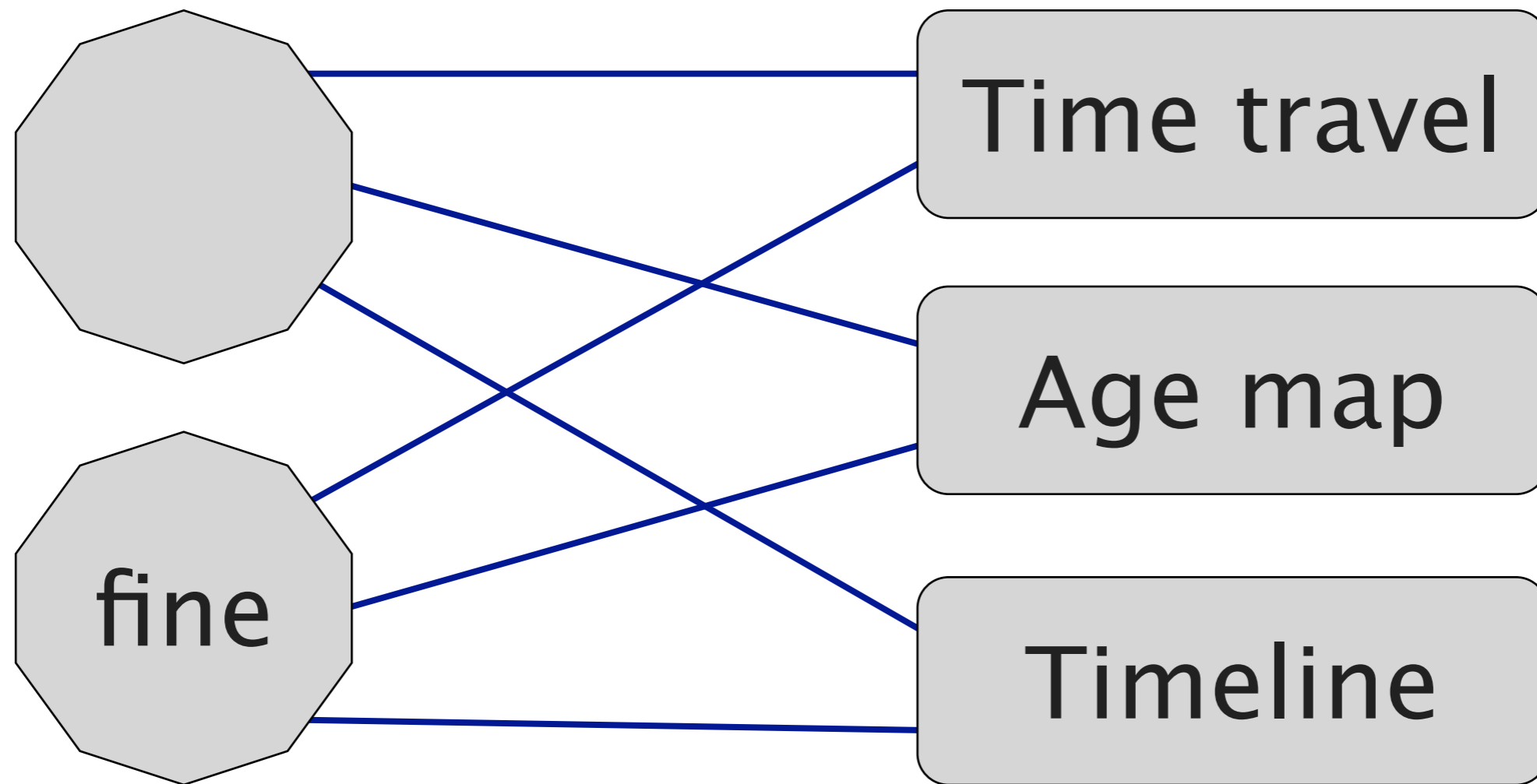


Time travel

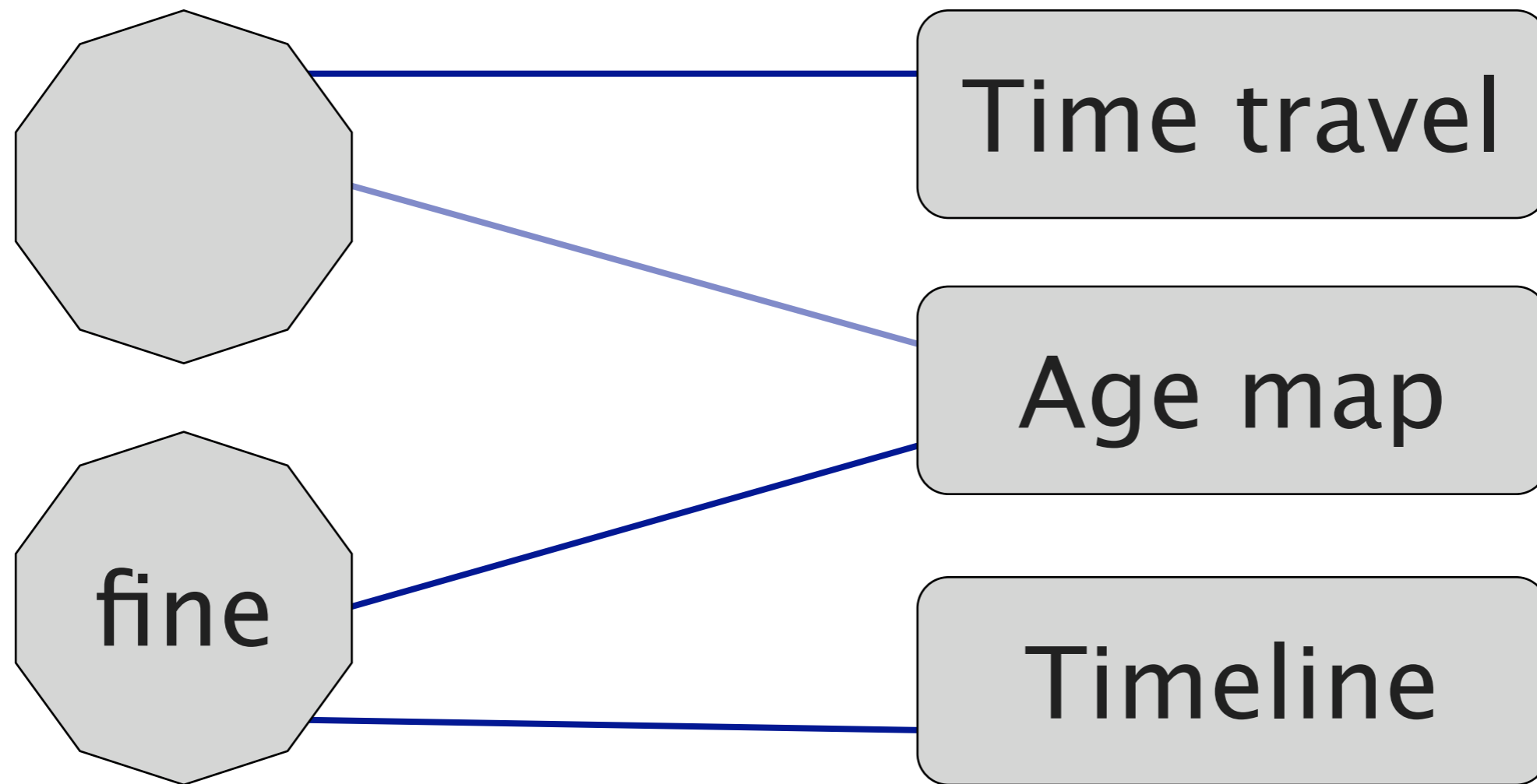
Age map

Timeline

Visualizing evolution

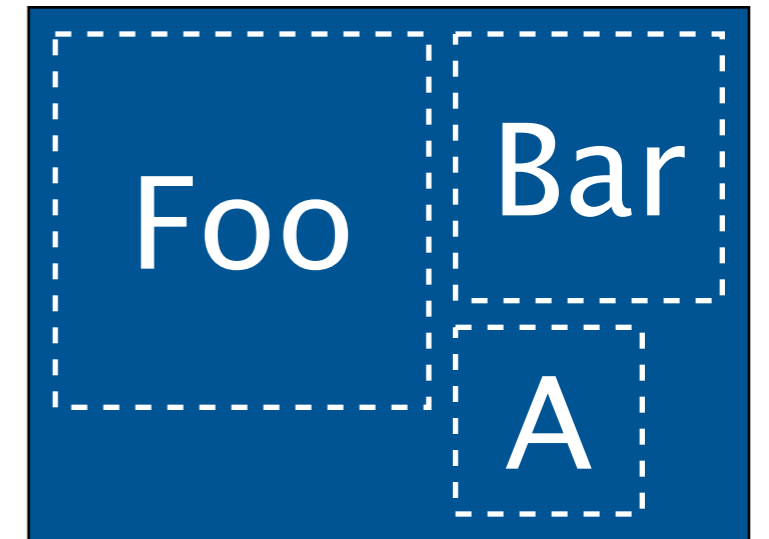


Visualizing evolution



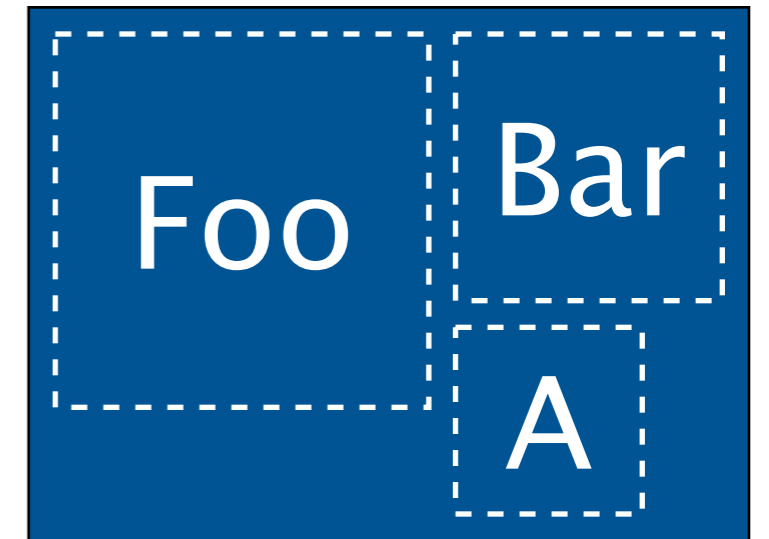
Consistent locality

Blueprint for the city,
based on the maximal
expansion of each artifact
(building, district).

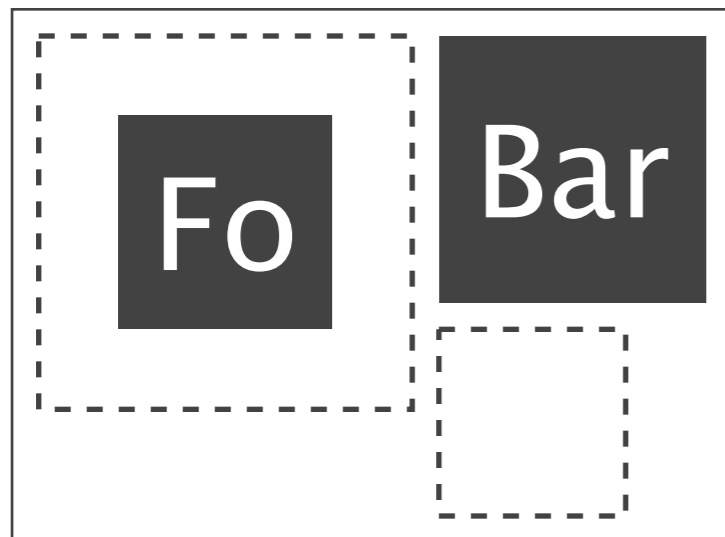


Consistent locality

Blueprint for the city,
based on the maximal
expansion of each artifact
(building, district).

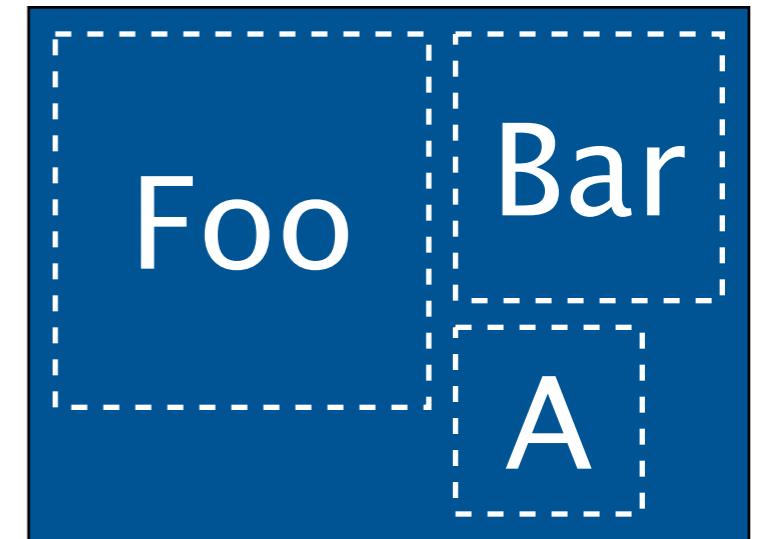


version 1

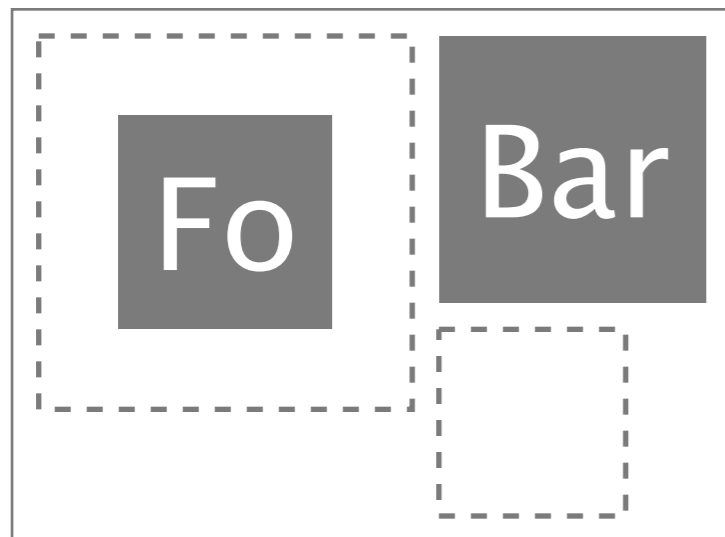


Consistent locality

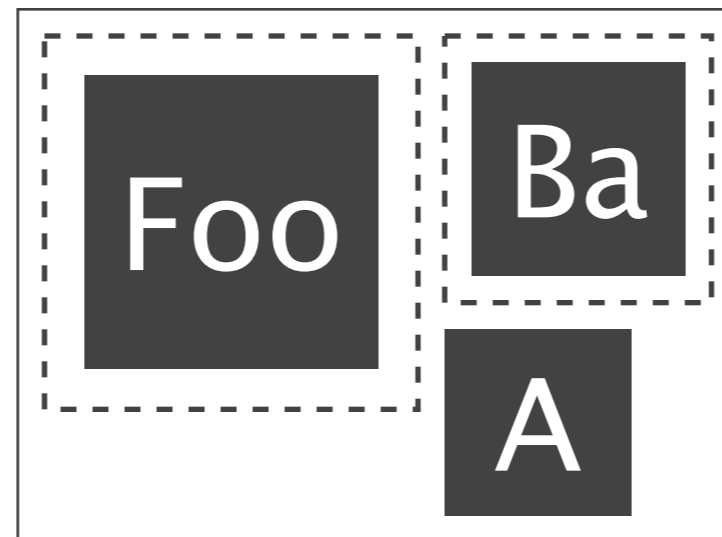
Blueprint for the city,
based on the maximal
expansion of each artifact
(building, district).



version 1

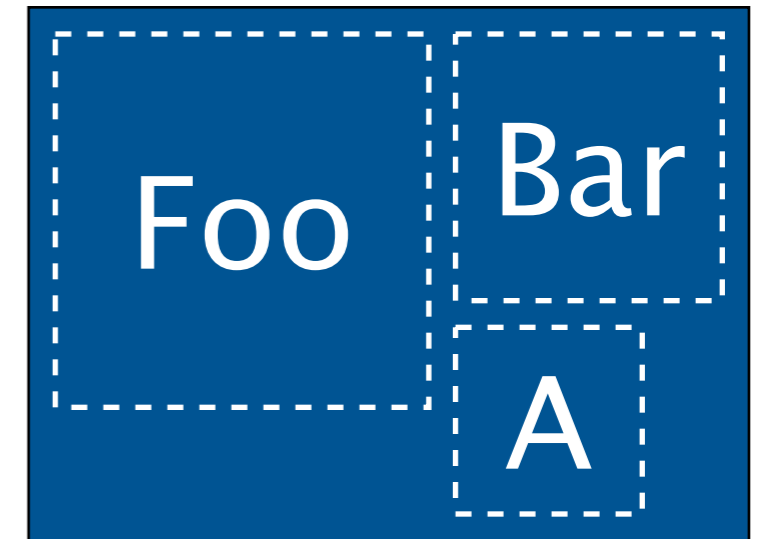


version 2

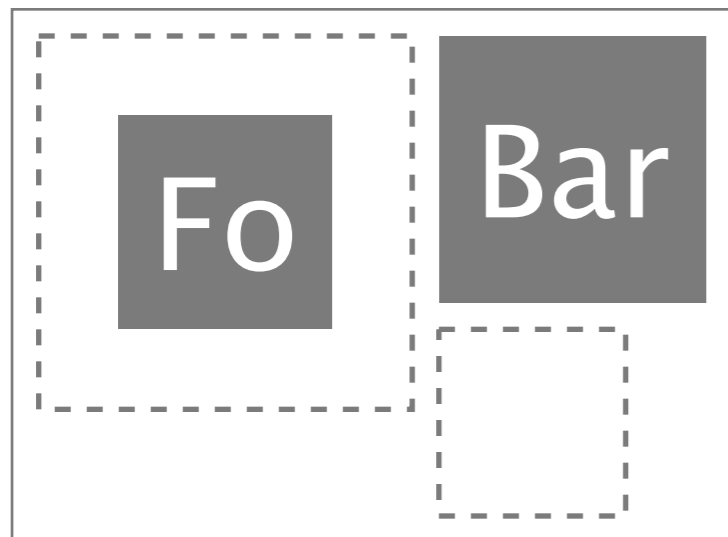


Consistent locality

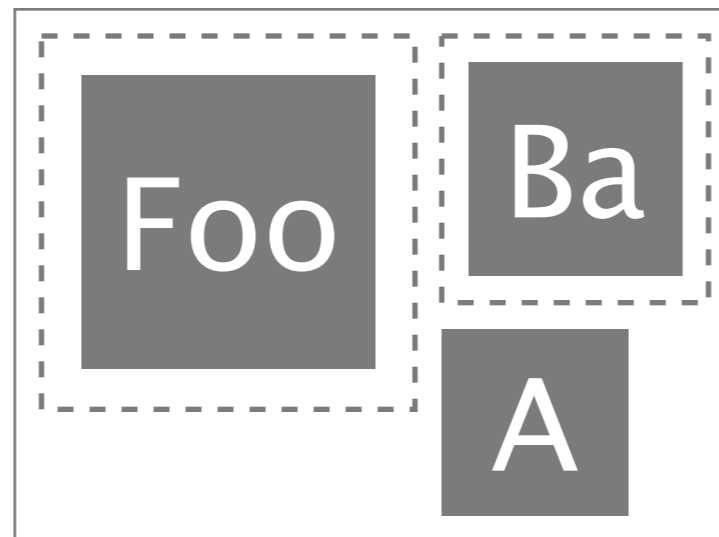
Blueprint for the city,
based on the maximal
expansion of each artifact
(building, district).



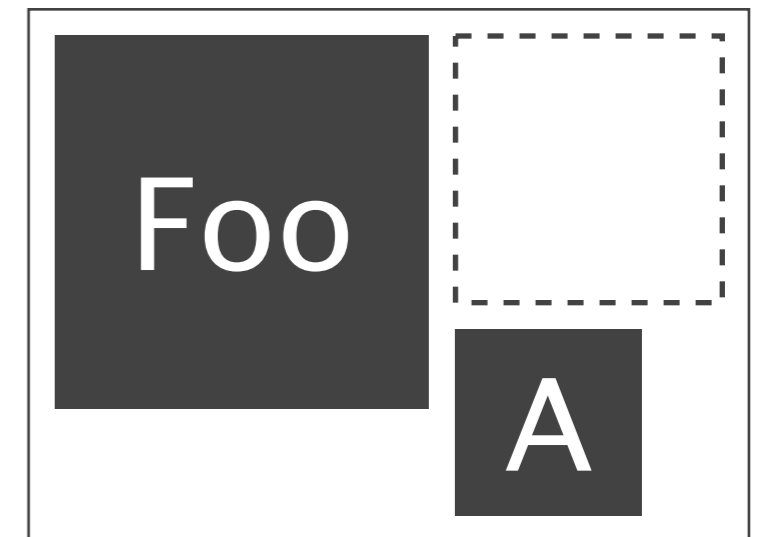
version 1



version 2



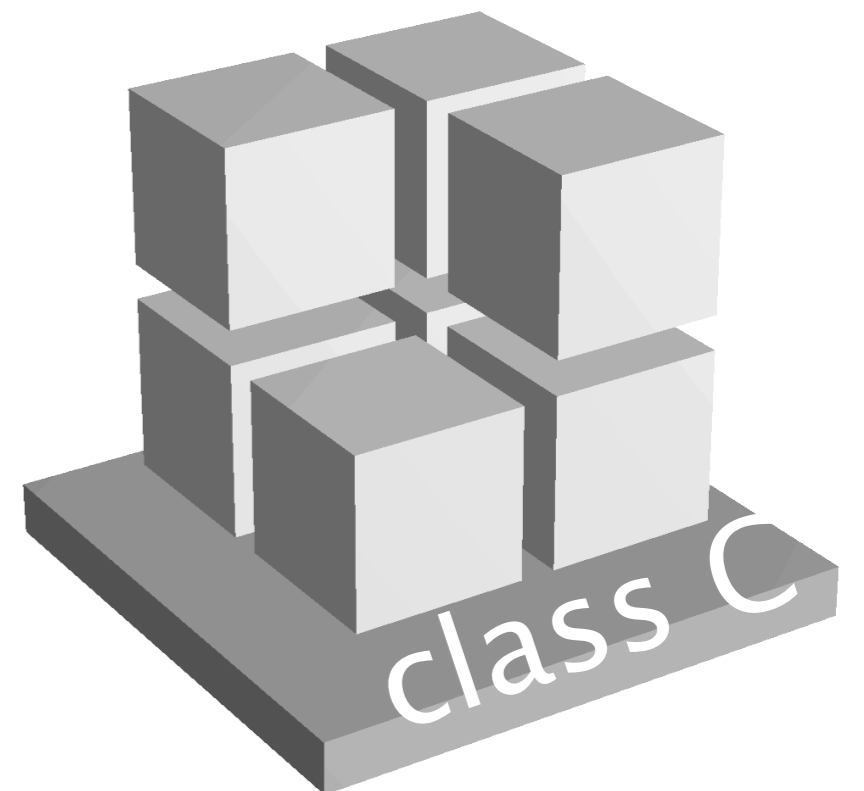
version 3



Age map

Bricks position is
chronological

neighbors are contemporary



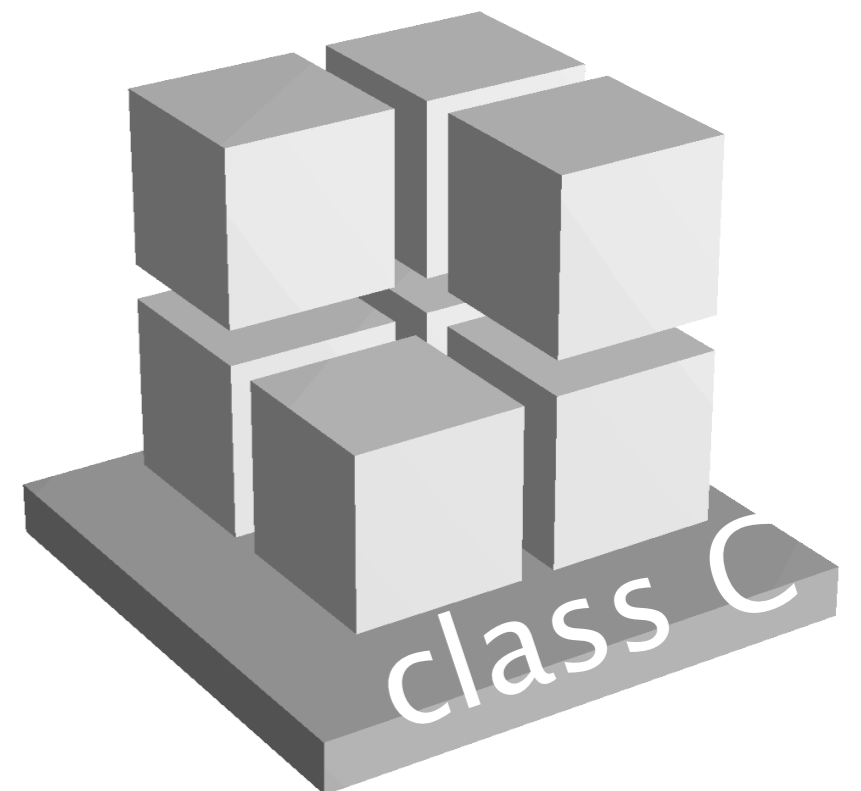
Age map

Superficial glimpse in the past.

color layer according to age (number of versions)

Bricks position is
chronological

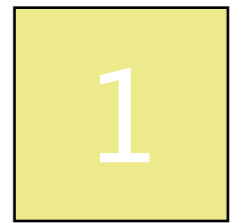
neighbors are contemporary



Age map

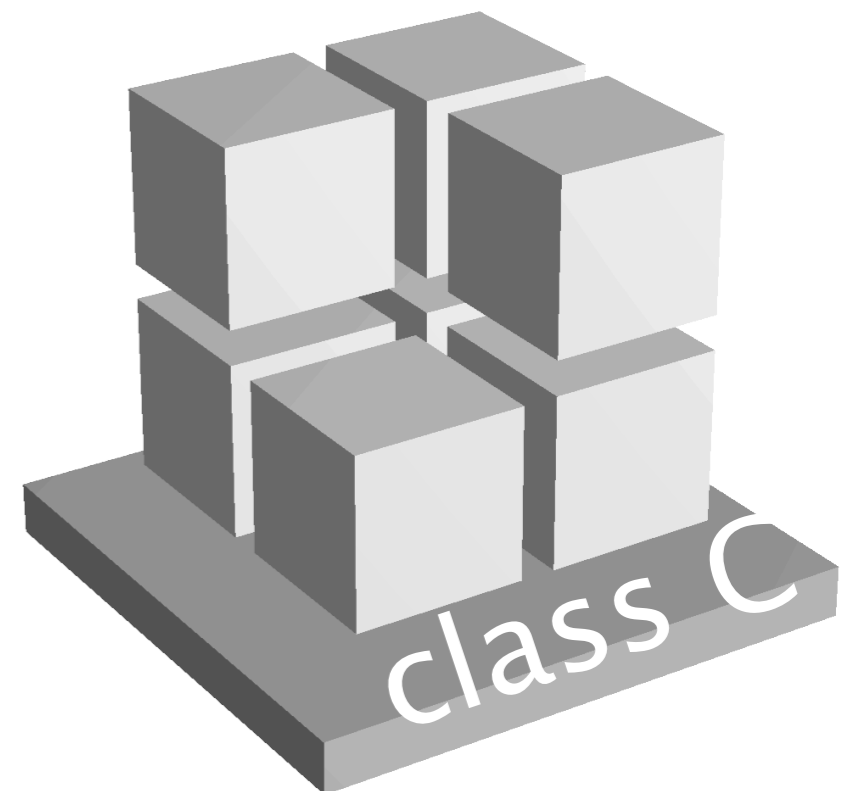
Superficial glimpse in the past.

color layer according to age (number of versions)



Bricks position is
chronological

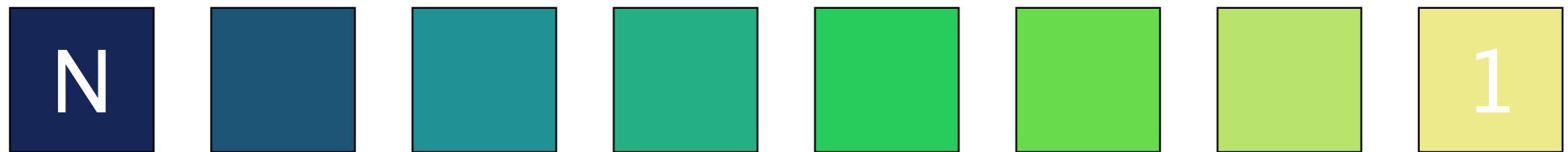
neighbors are contemporary



Age map

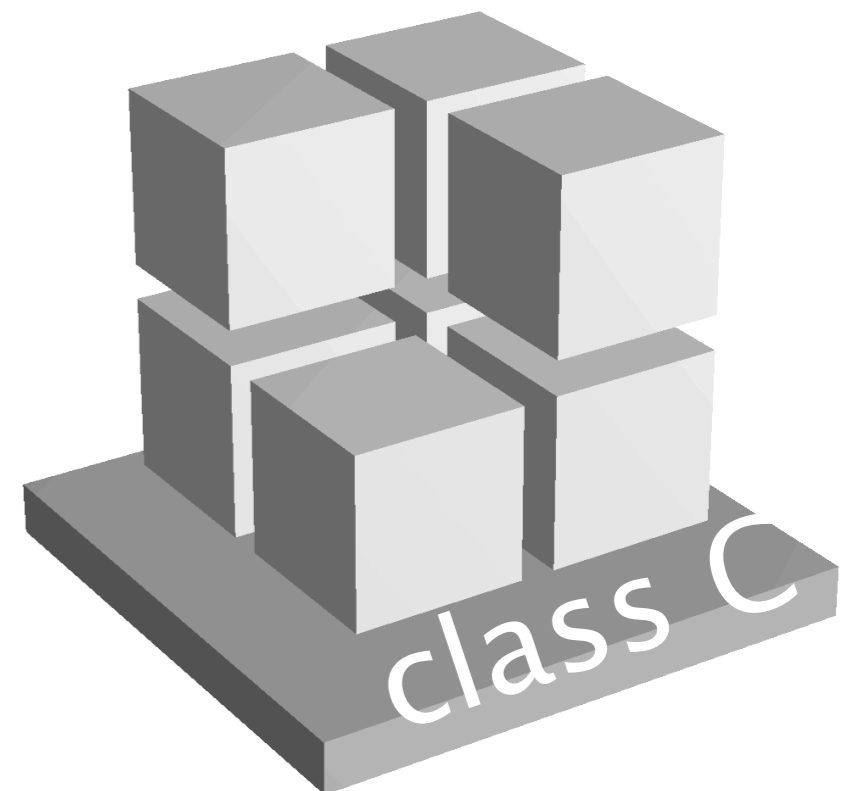
Superficial glimpse in the past.

color layer according to age (number of versions)



Bricks position is
chronological

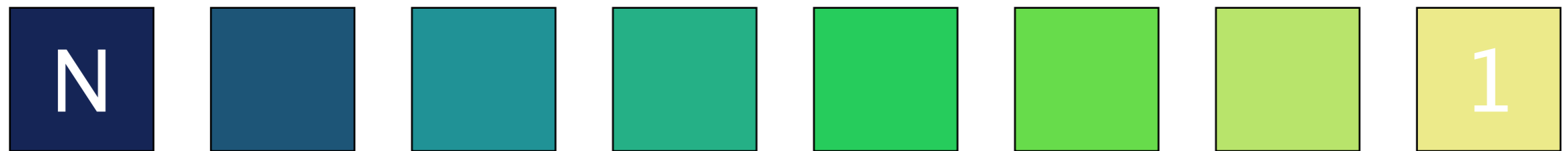
neighbors are contemporary



Age map

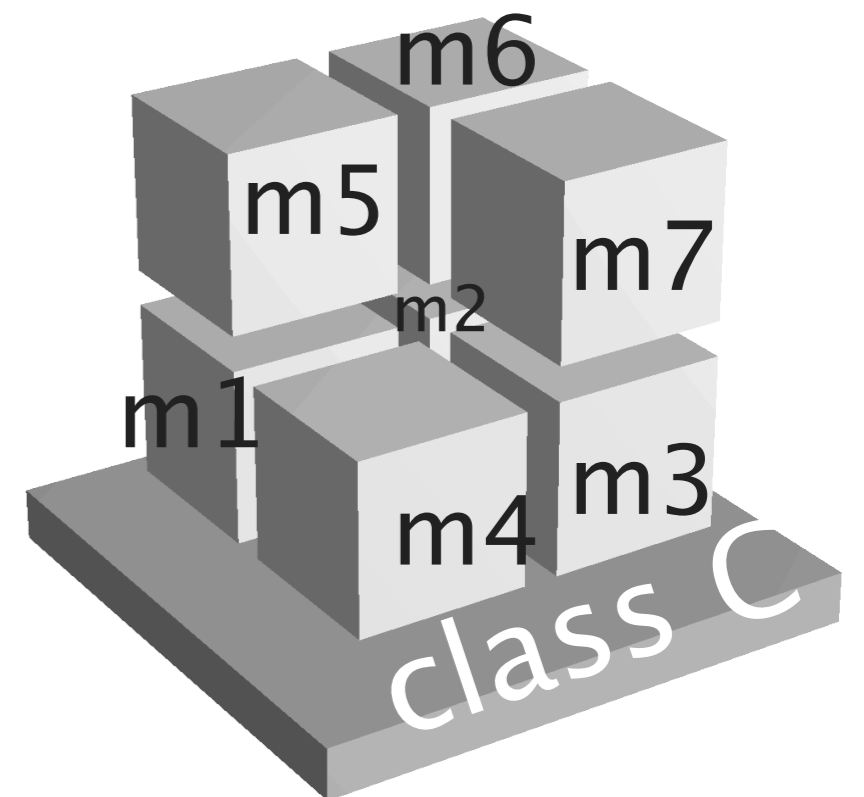
Superficial glimpse in the past.

color layer according to age (number of versions)



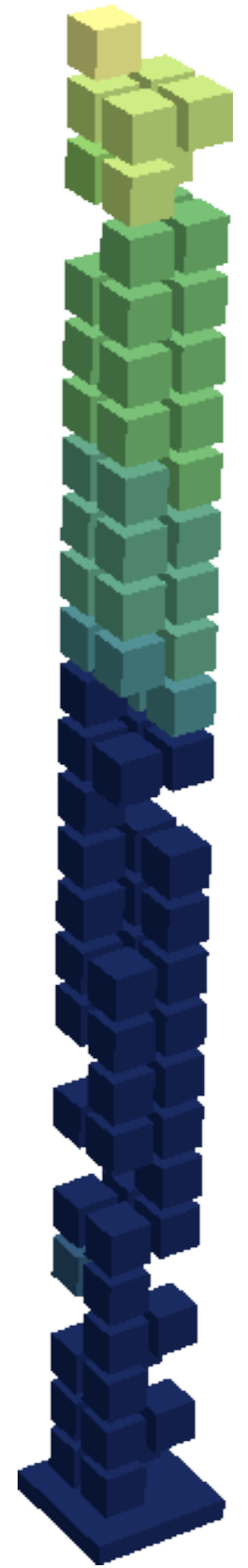
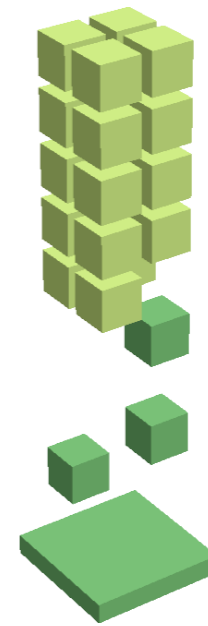
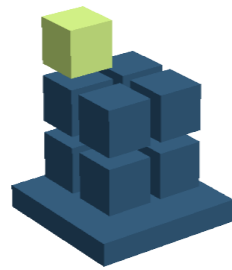
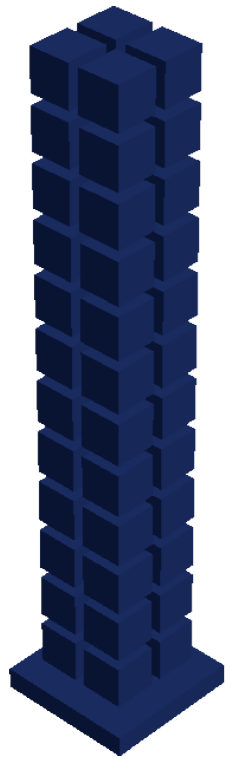
Bricks position is
chronological

neighbors are contemporary

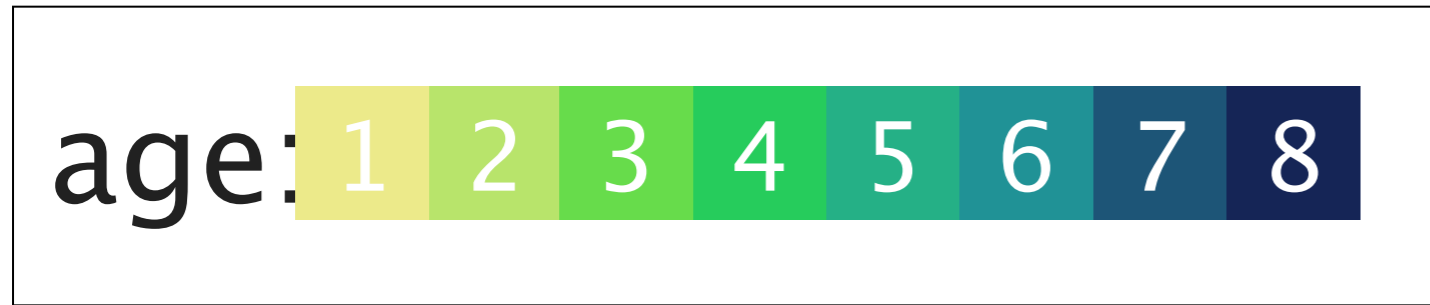


Age map interpretation

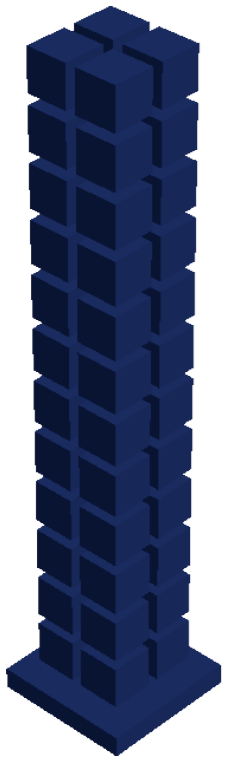
age: 1 2 3 4 5 6 7 8



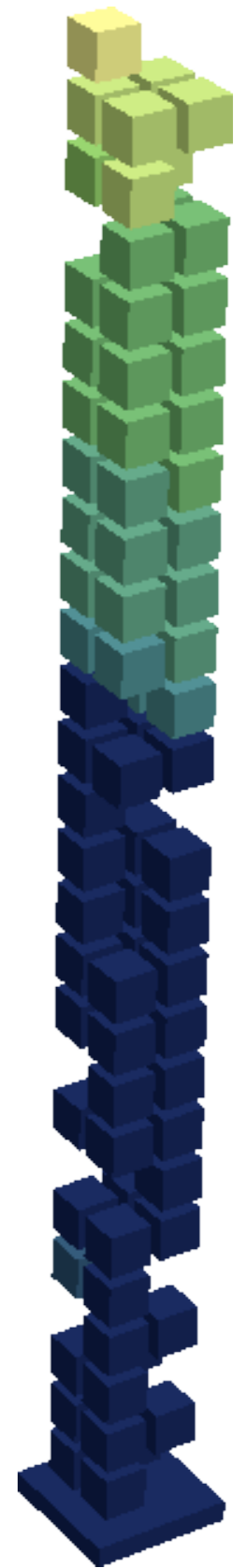
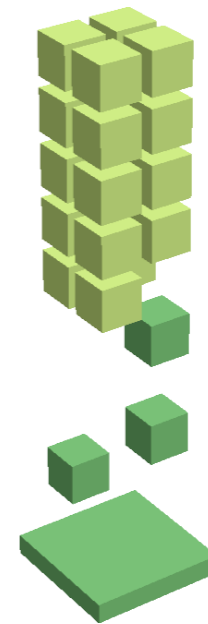
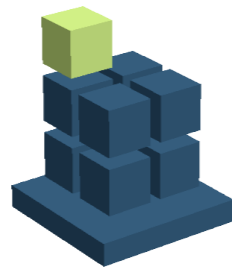
Age map interpretation



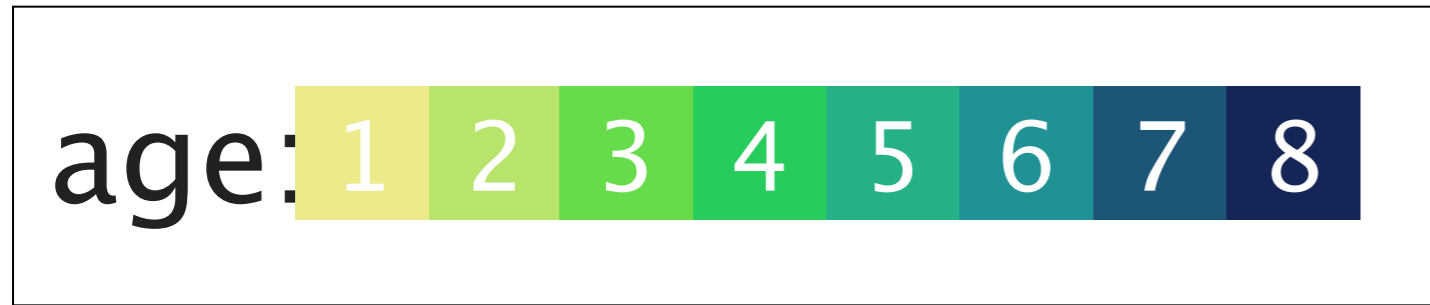
stable



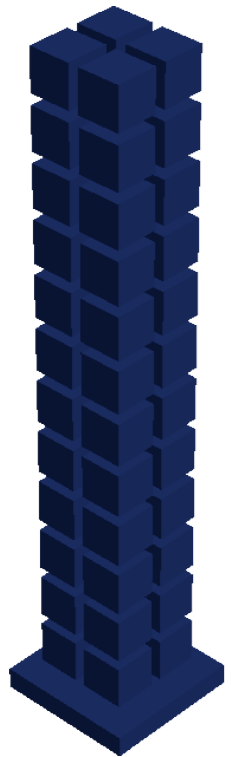
very old



Age map interpretation

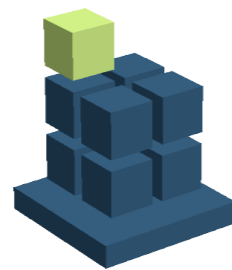


stable

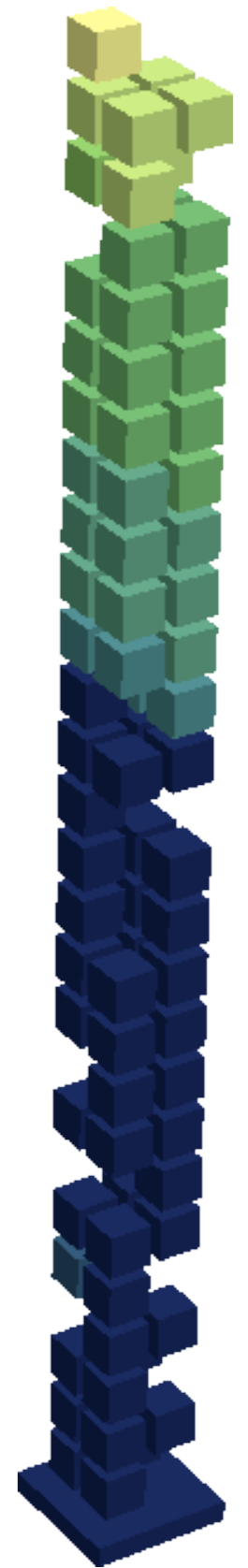
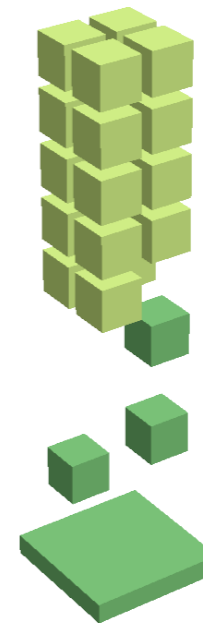


very old

rarely updated



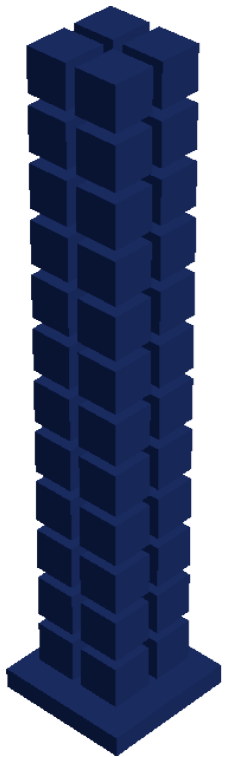
old



Age map interpretation

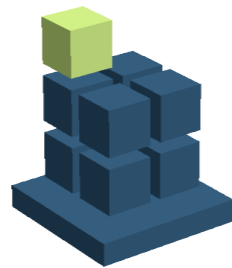
age: 1 2 3 4 5 6 7 8

stable



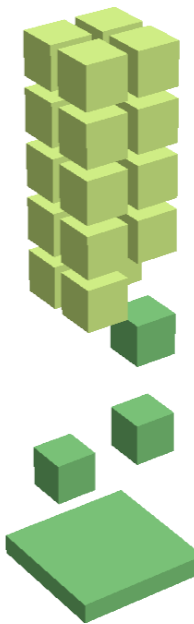
very old

rarely updated

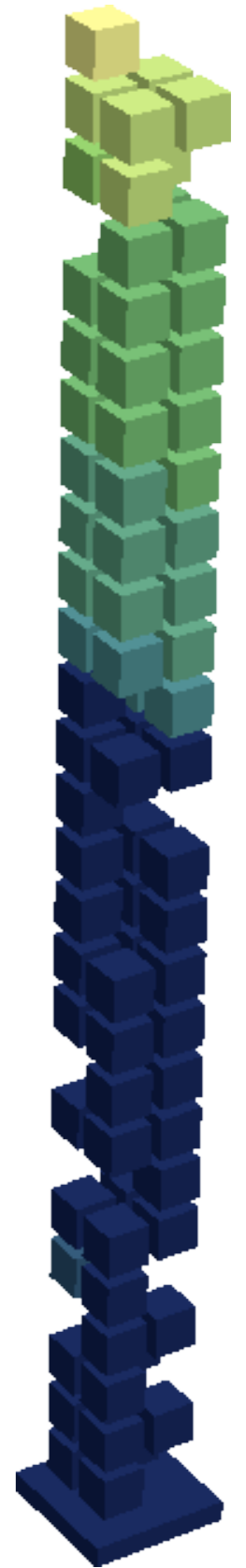


old

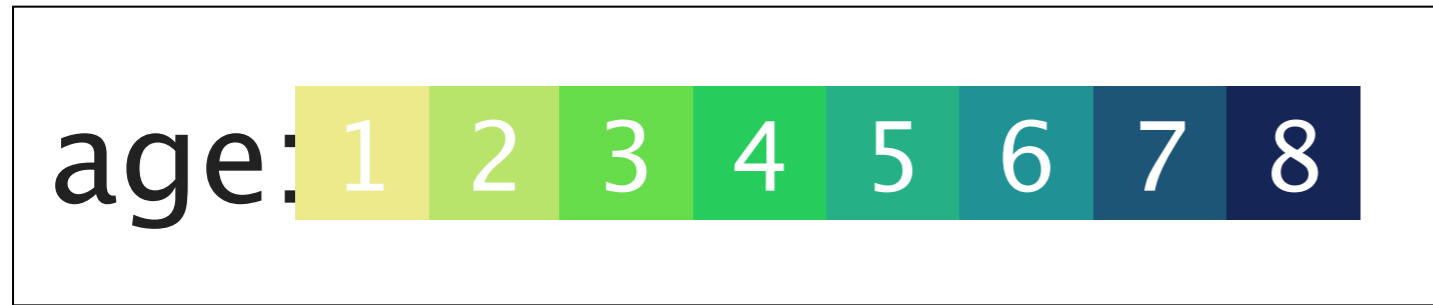
highly unstable



young

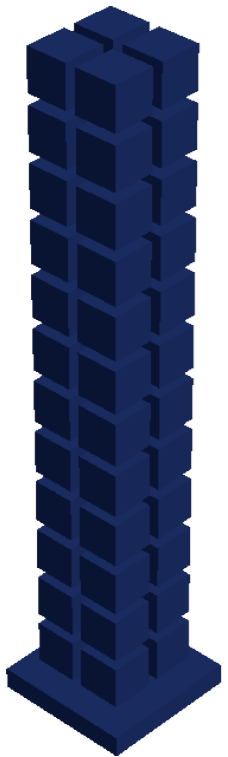


Age map interpretation



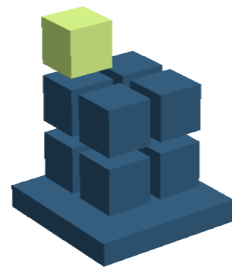
updated
often
rather
unstable

stable



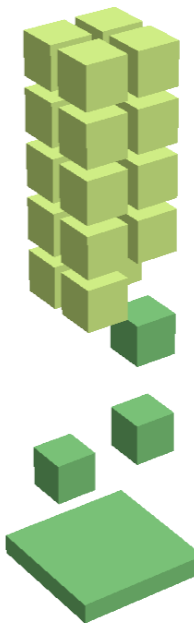
very old

rarely updated

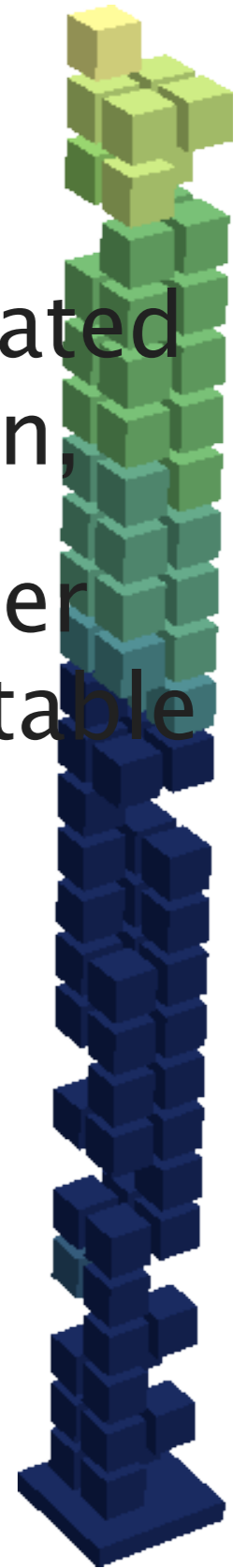


old

highly unstable

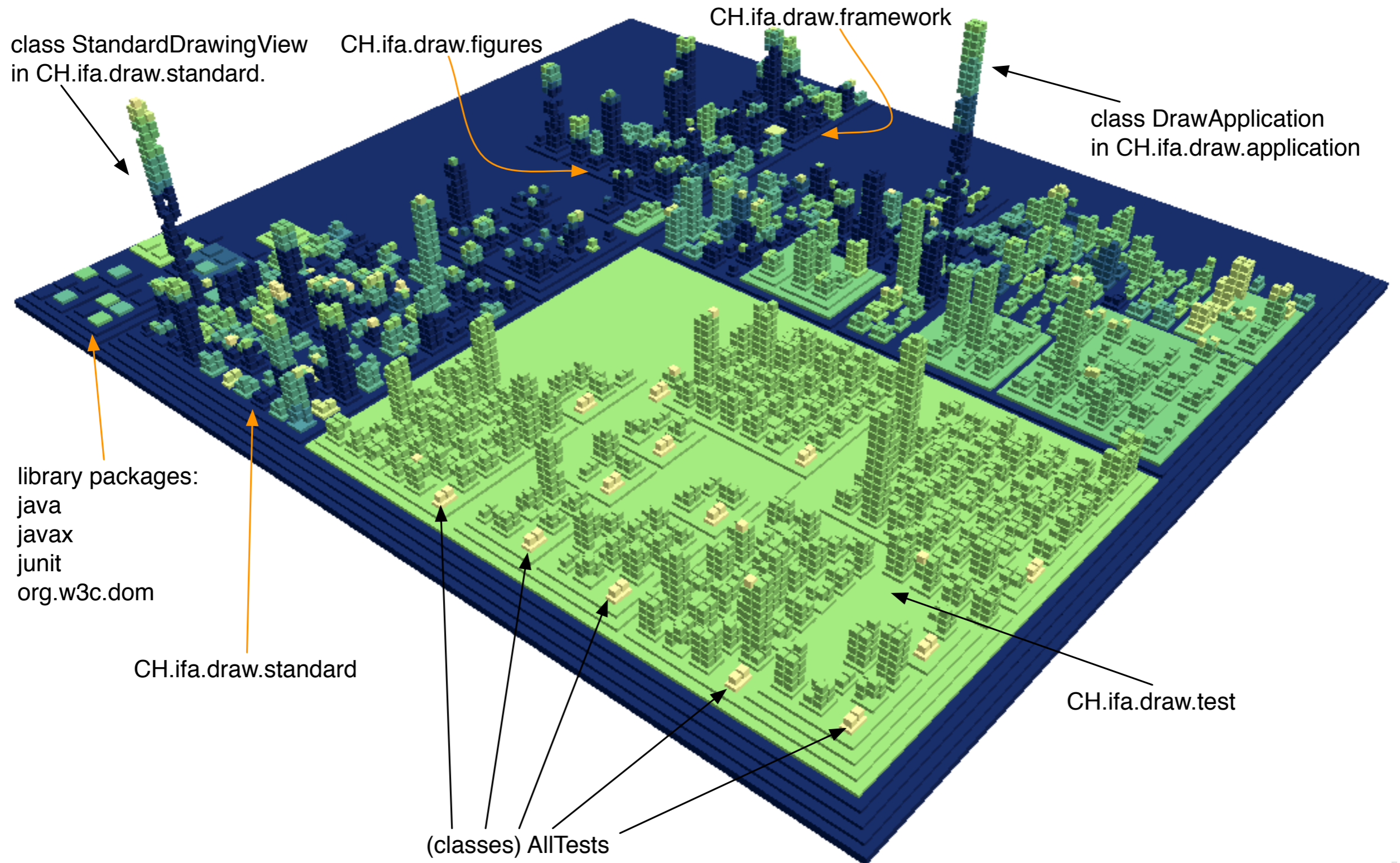


young



very old

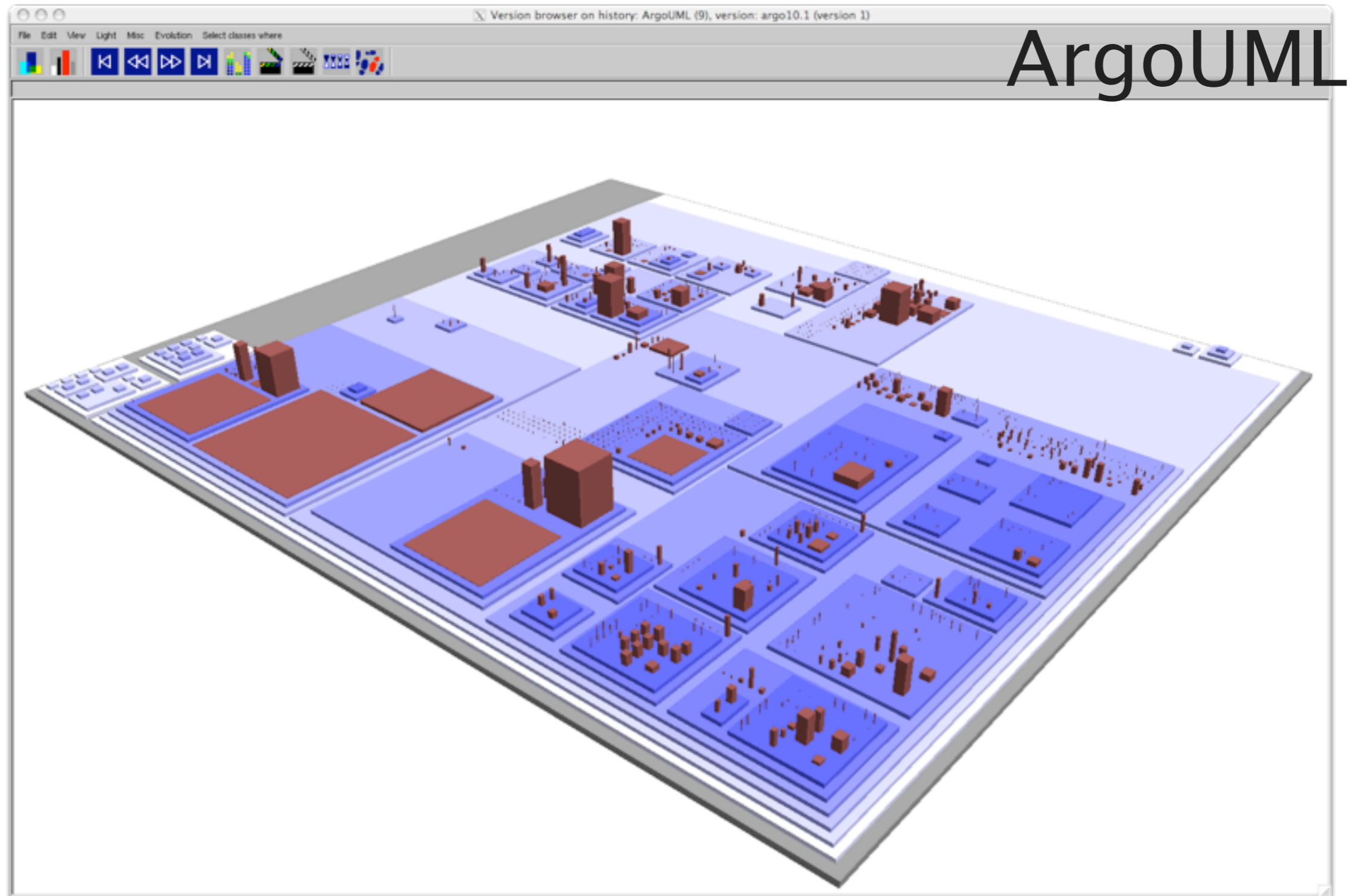
Age map, fine-grained



Time travel, coarse-grained

ArgoUML

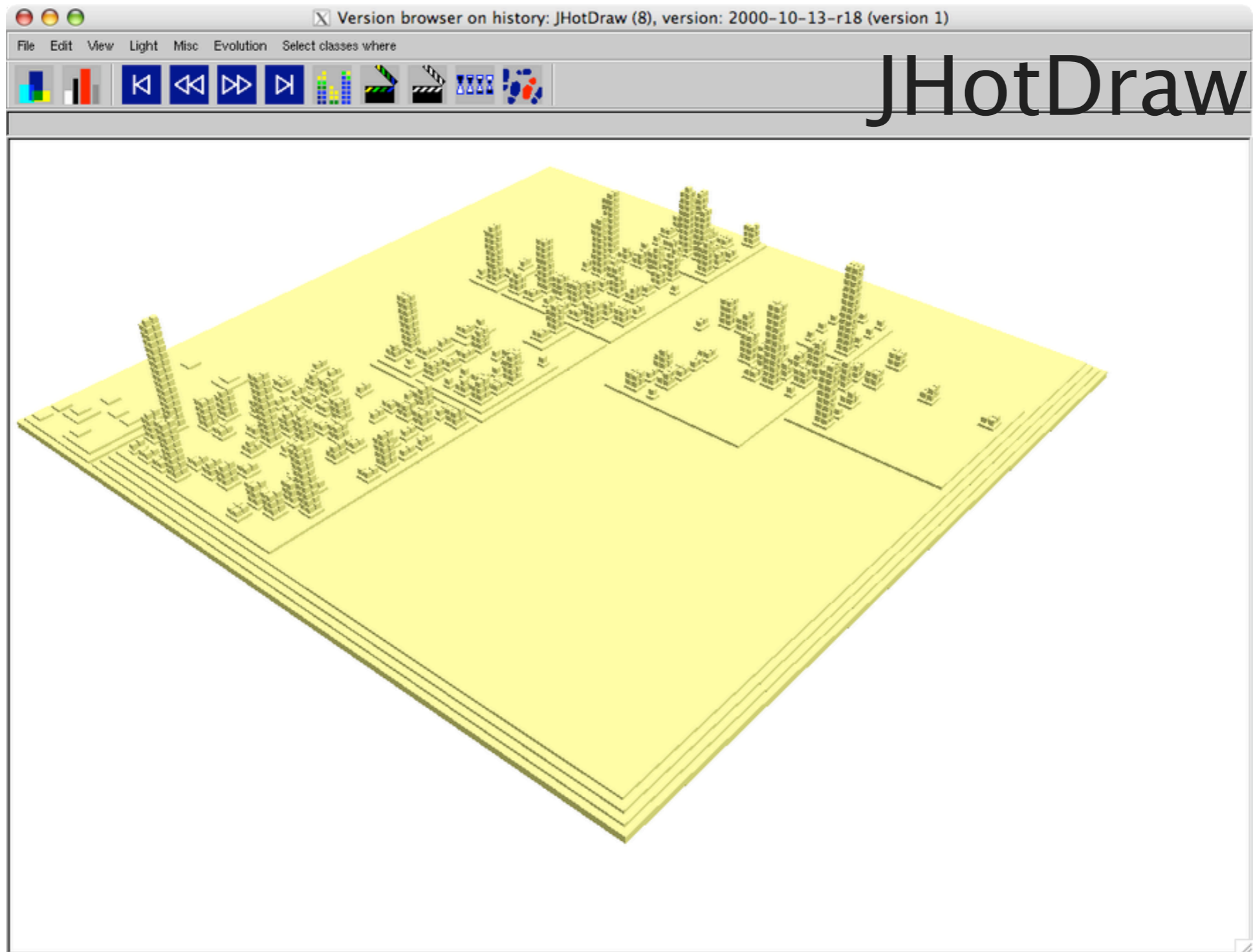
Time travel, coarse-grained



Time travel, fine-grained

JHotDraw

Time travel, fine-grained

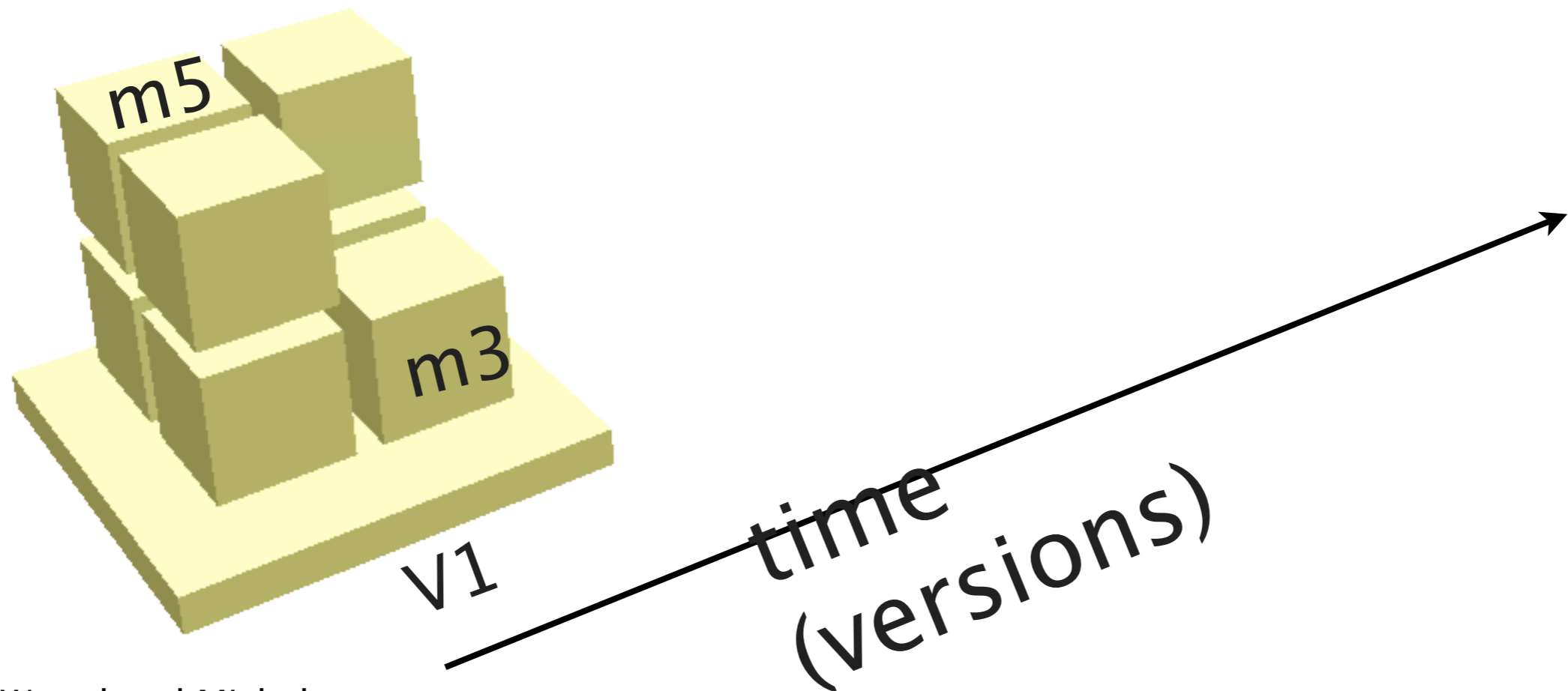


Timeline

History of class C

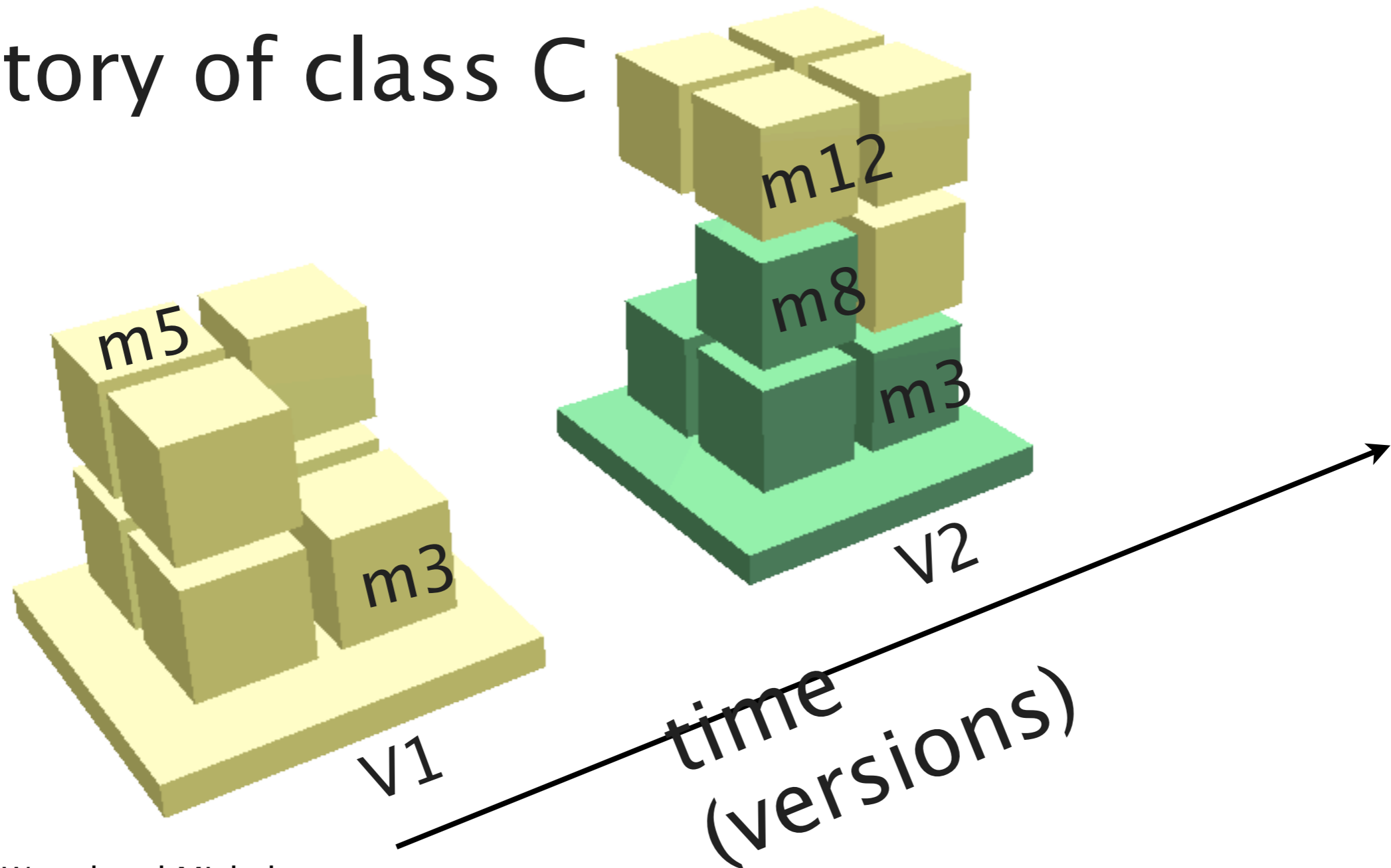
Timeline

History of class C



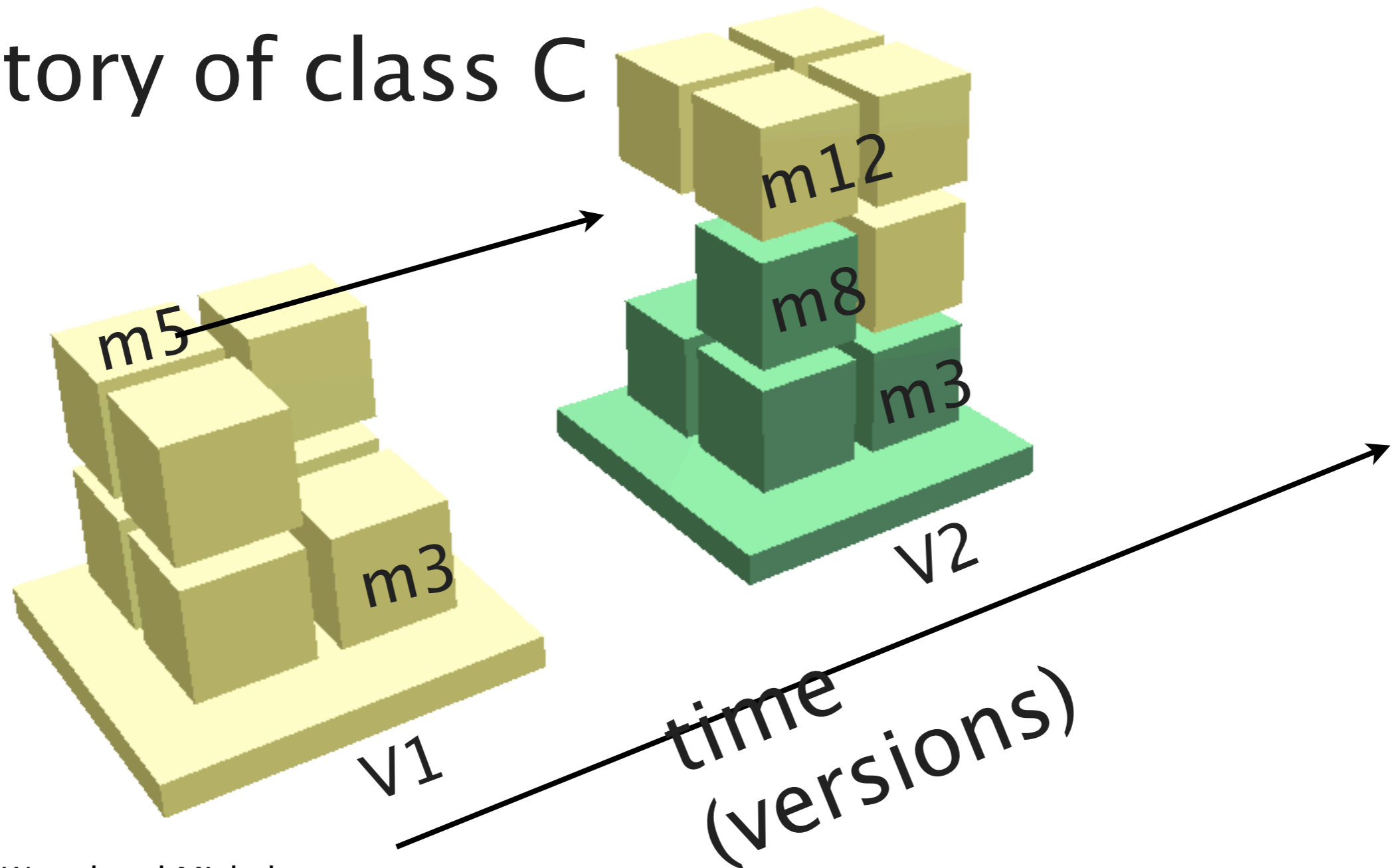
Timeline

History of class C



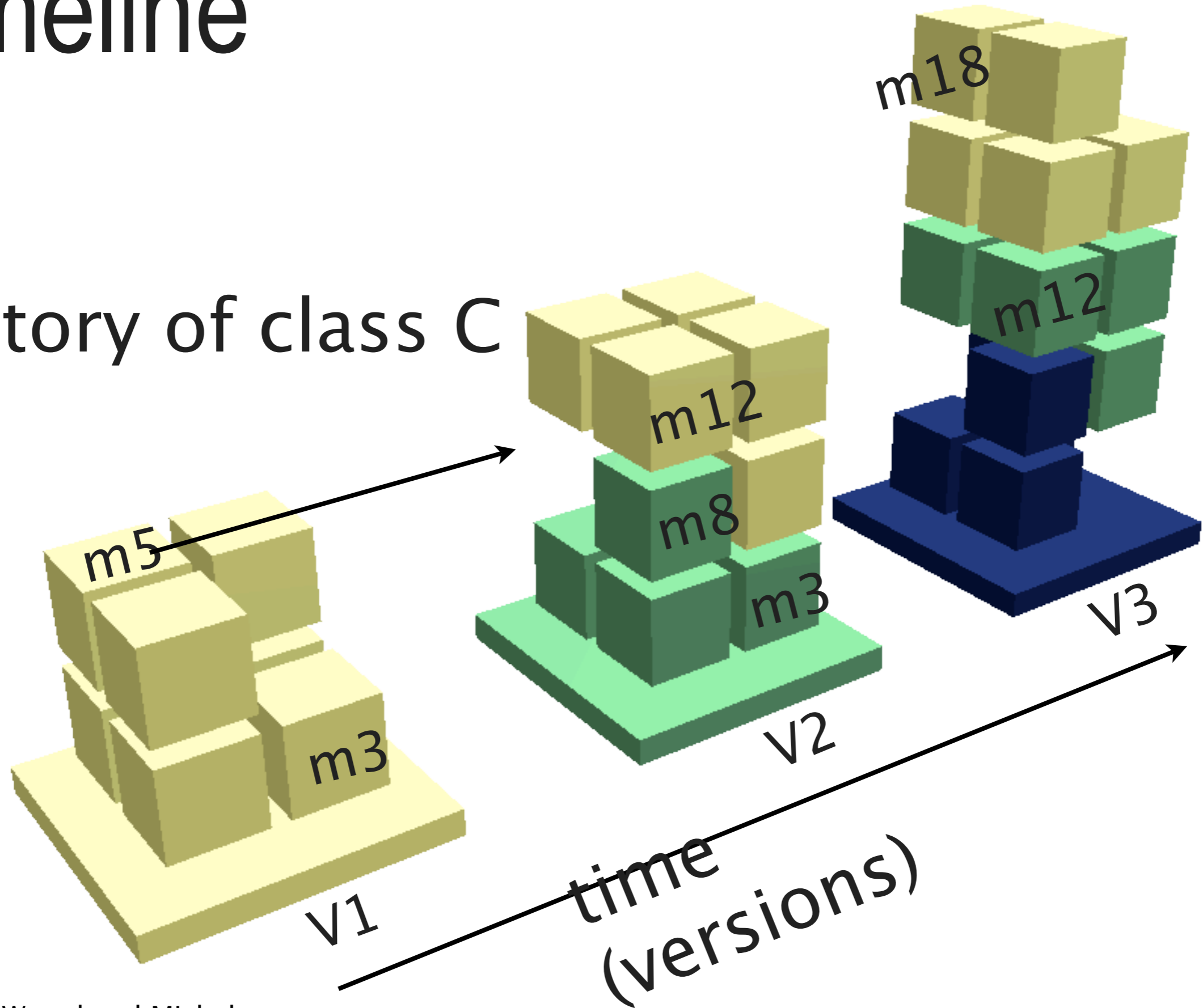
Timeline

History of class C



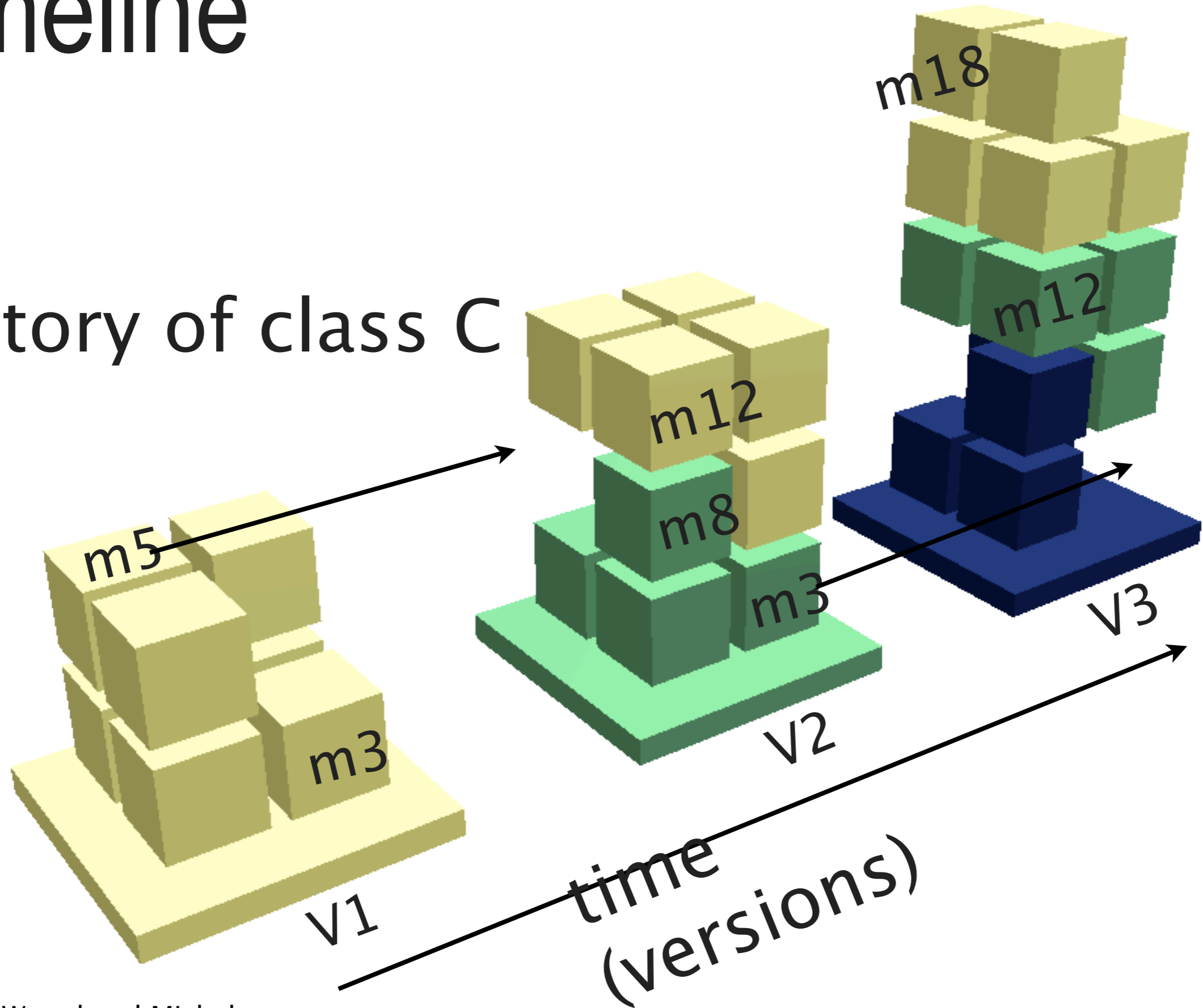
Timeline

History of class C

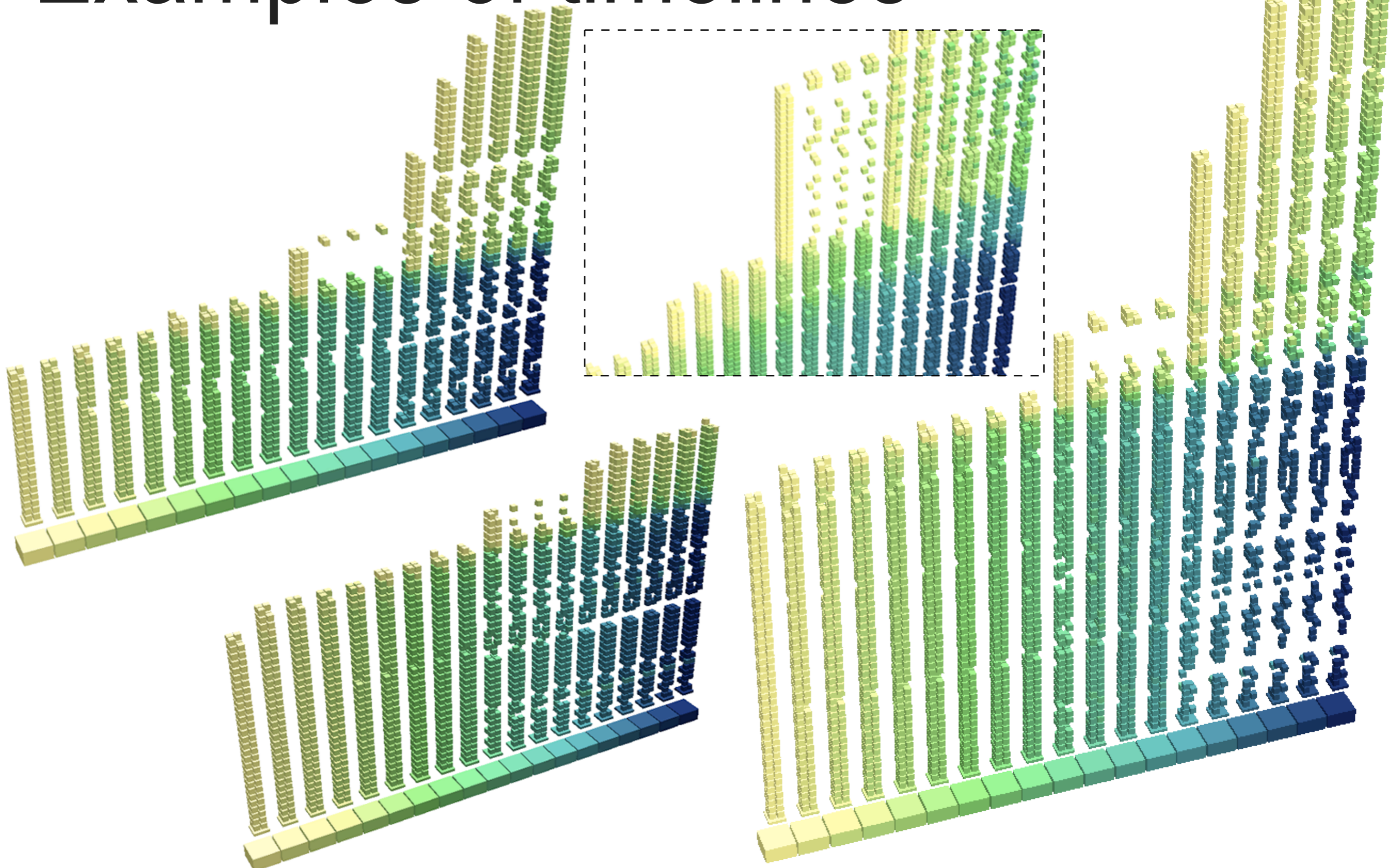


Timeline

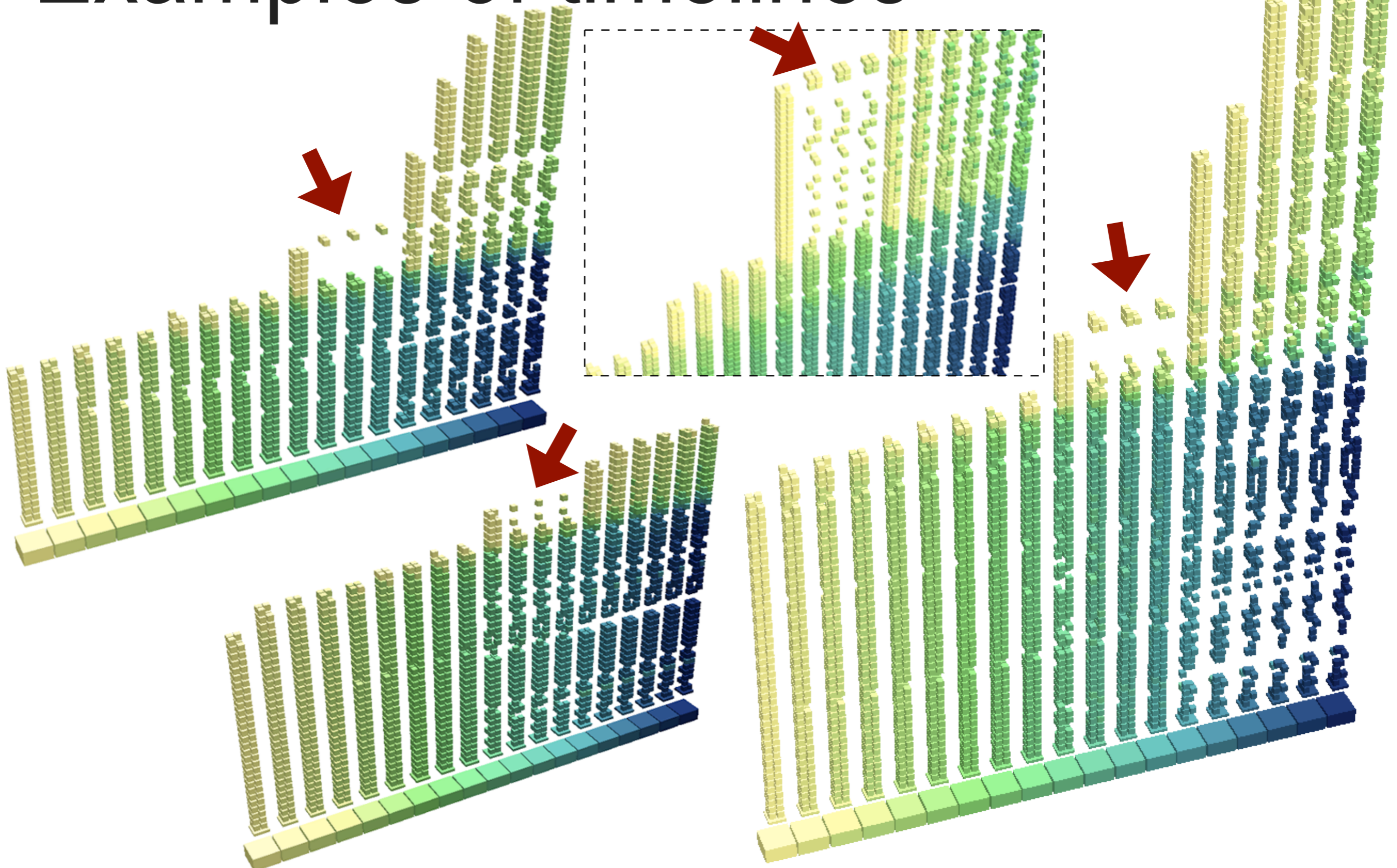
History of class C



Examples of timelines



Examples of timelines



Scripting cities

The screenshot shows the CodeCity Easel application window. The main view displays a 3D city model composed of numerous red brick buildings connected by a dense network of blue lines representing inheritance relationships. The interface includes a top status bar with 'Selected: 0 figures', a right-hand panel for object inspection, and a bottom panel for scripting.

Model
name:
Root::Smalltalk::Core::Object.myEventTable:(Object)
type: FAMIXMethod

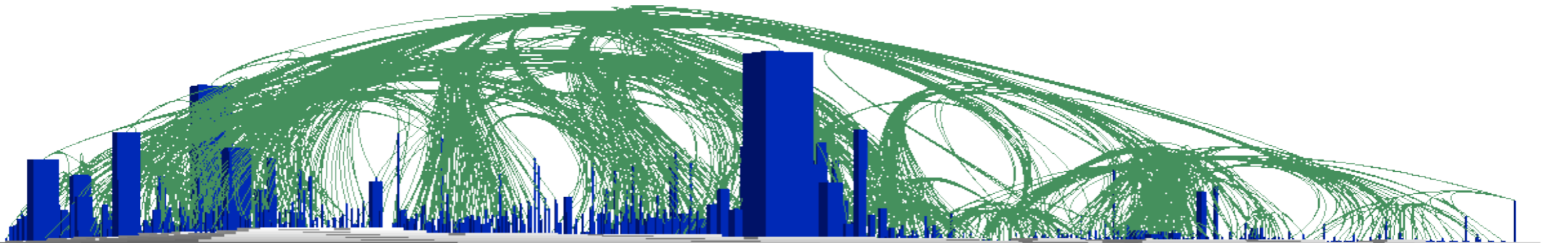
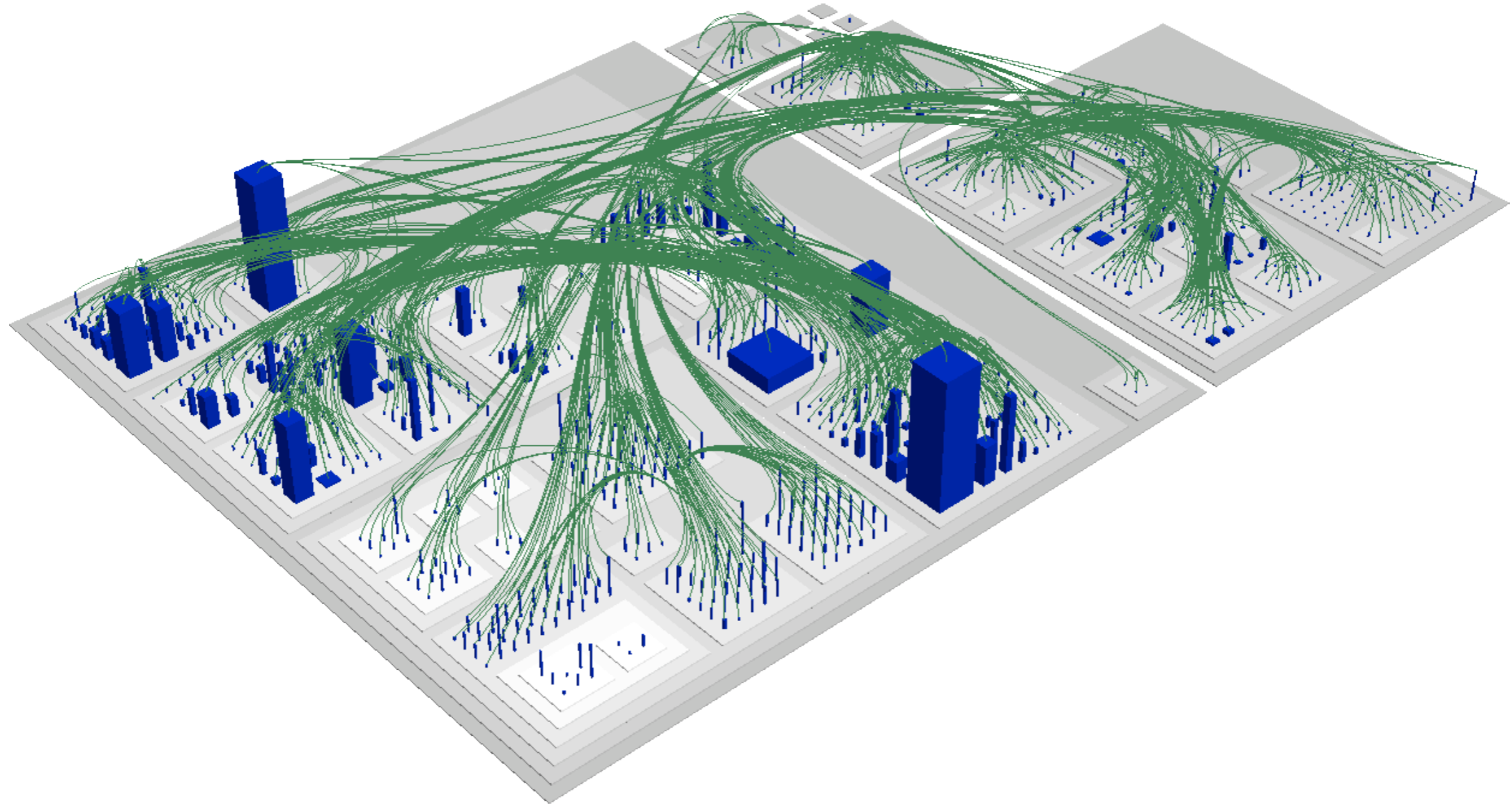
Figure
type: cuboid
alpha: 1(block evaluation)
color: CodeFoo.Color darkRed(block evaluation)
position: (1.6,111.4,15.8)
width: 1
length: 1
height: 1

Variable	Object
mooseModel	MooseDevelopment

```
view nodeShape  
  glyphClass: TerrainGlyph;  
  mapper: (ConstantMapper new constantValue: 0.2) on: #terrainThickness;  
  layout: (ProgressiveBricksLayout new horizontalGap: 0.2; verticalGap: 0.2).  
view nodes: (mooseModel allClasses select: [:e| e numberOfMethods > 5]) forEach: [:eachClass]  
  view nodeShape  
    alpha: [:e| eachClass isMetaclass ifTrue: [0.5] ifFalse: [1]];  
    color: [:e|Color darkRed].  
  view nodes: eachClass methods].  
view edgeShape color: [:e|Color navy].  
view edges: mooseModel allInheritanceDefinitions from: #superclass to: #subclass
```

Refresh View
Clear View

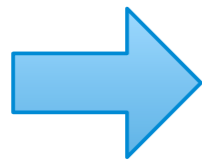
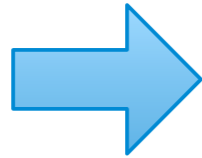
On relationships





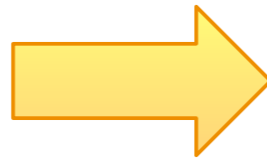
The tool

Tool chain

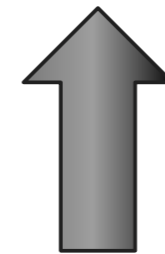
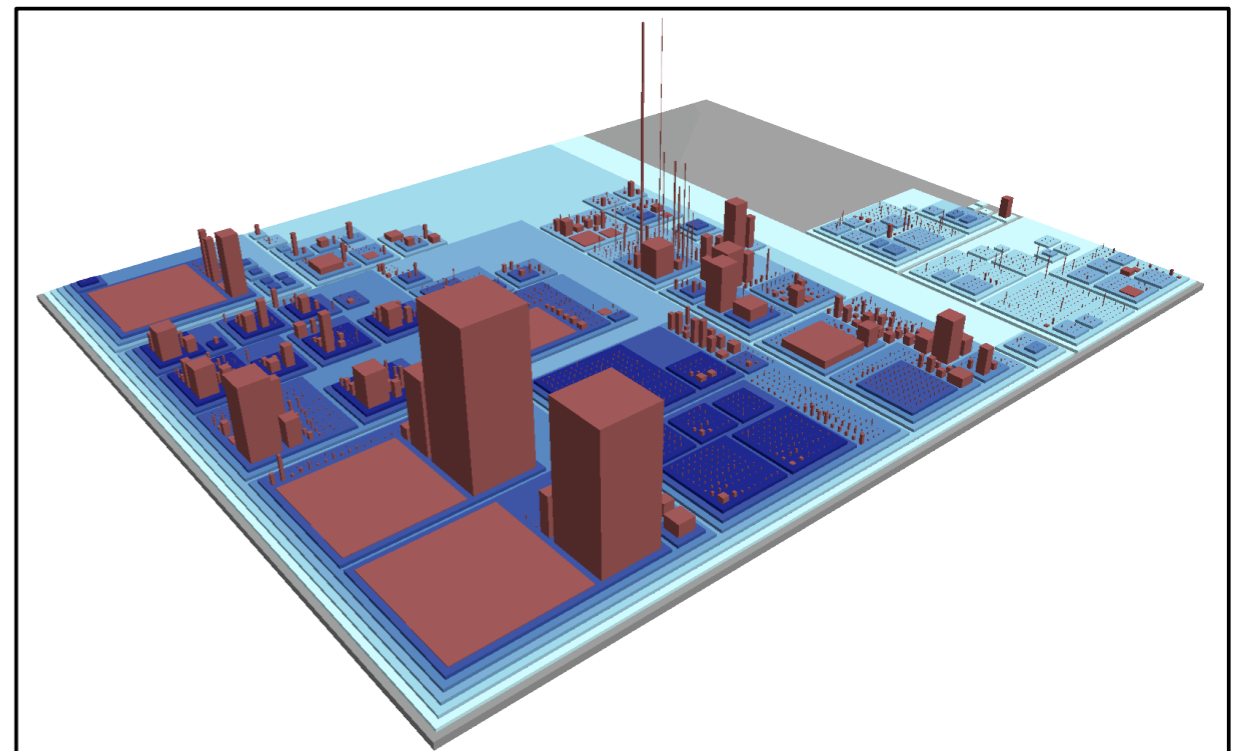


iPlasma

model
exchange



Moose



Jun

rendering

CodeCity

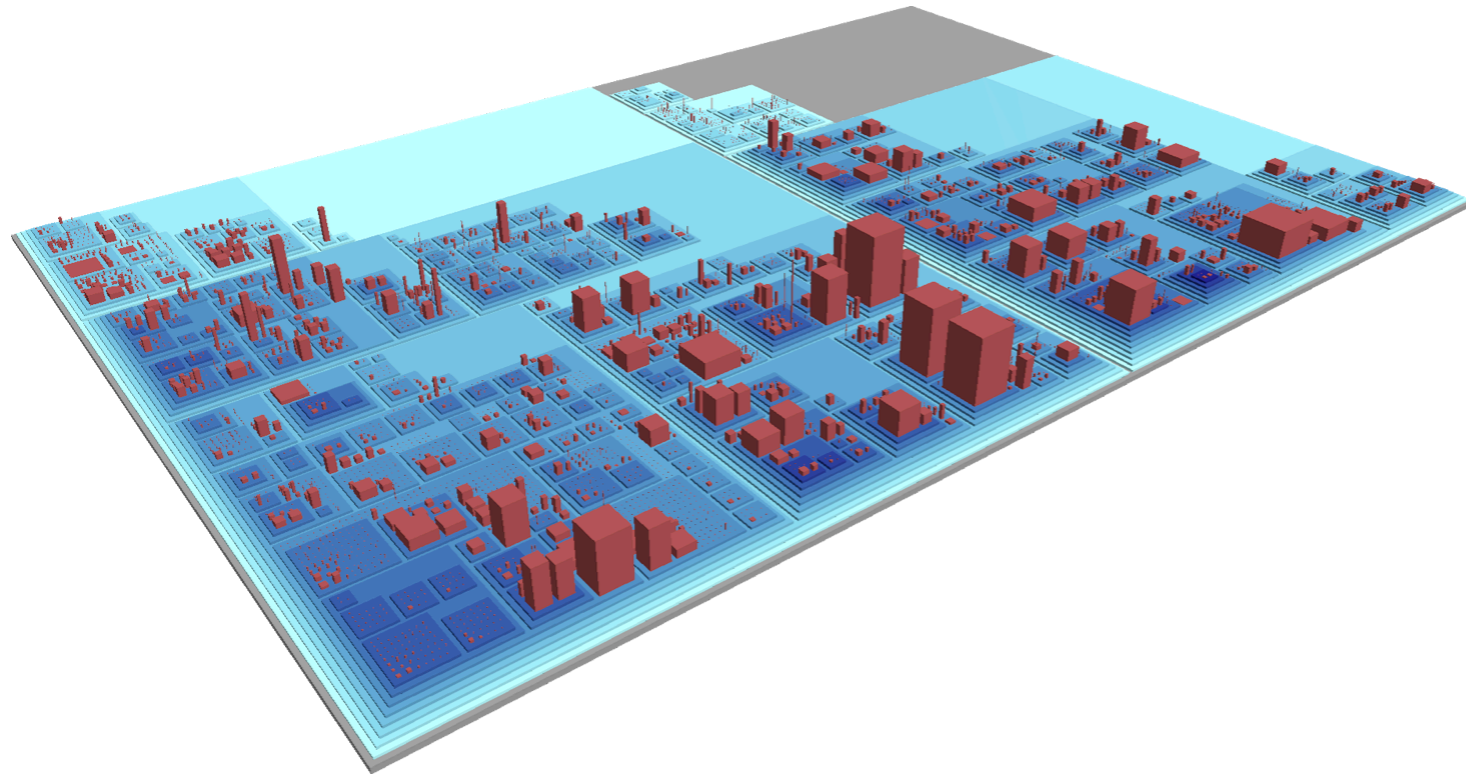
VisualWorks 7.6

Cityscapes

Cityscapes

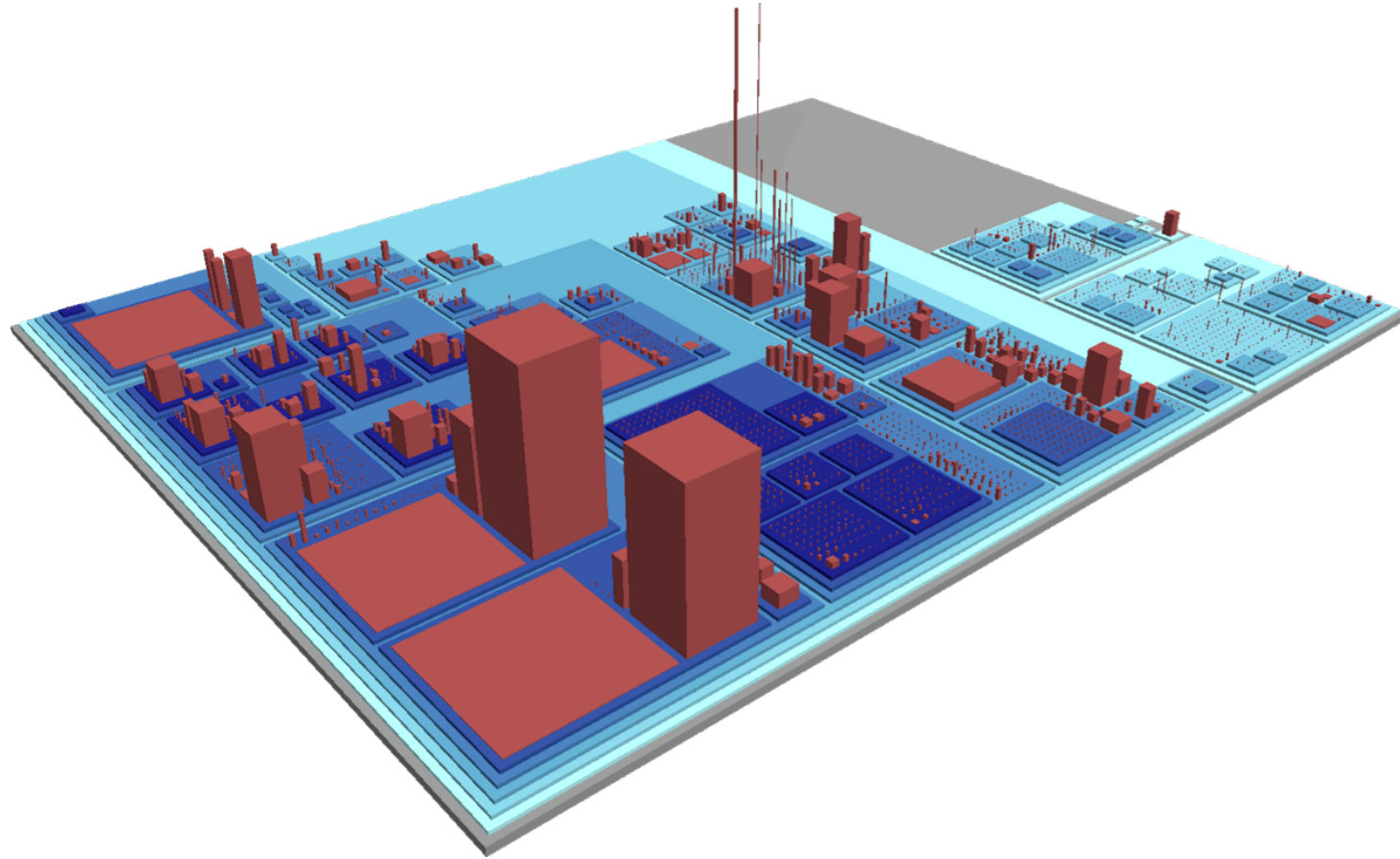
System	Languag	NOP	NOC	kLO

Cityscapes



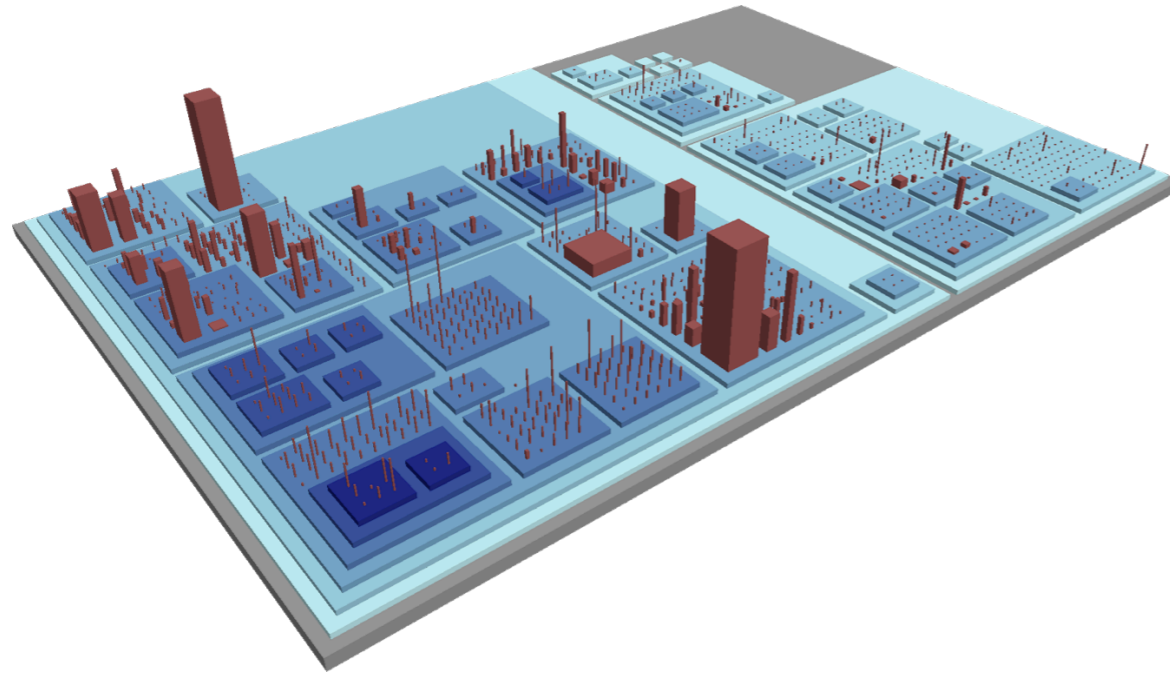
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274

Cityscapes



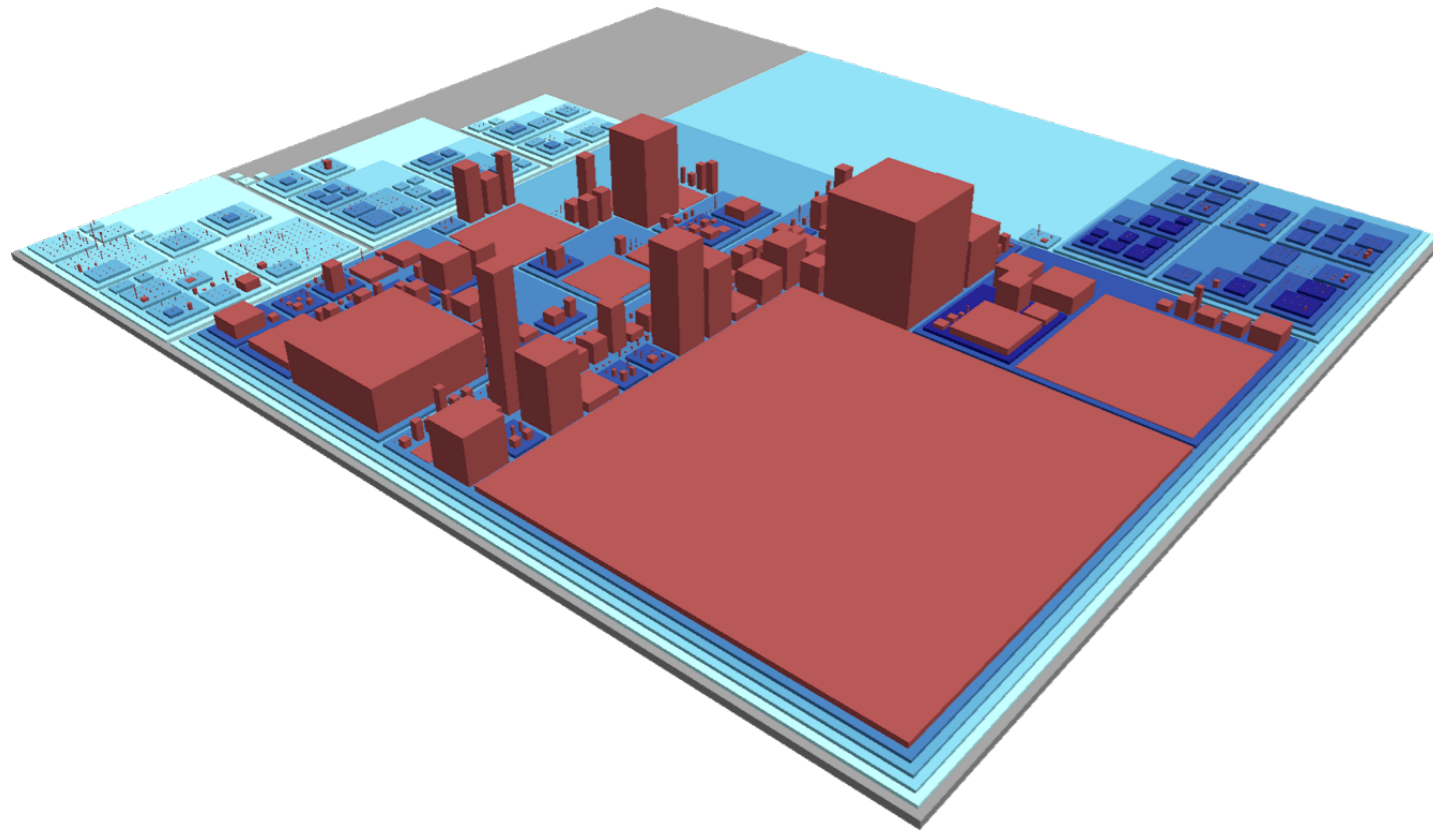
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137

Cityscapes



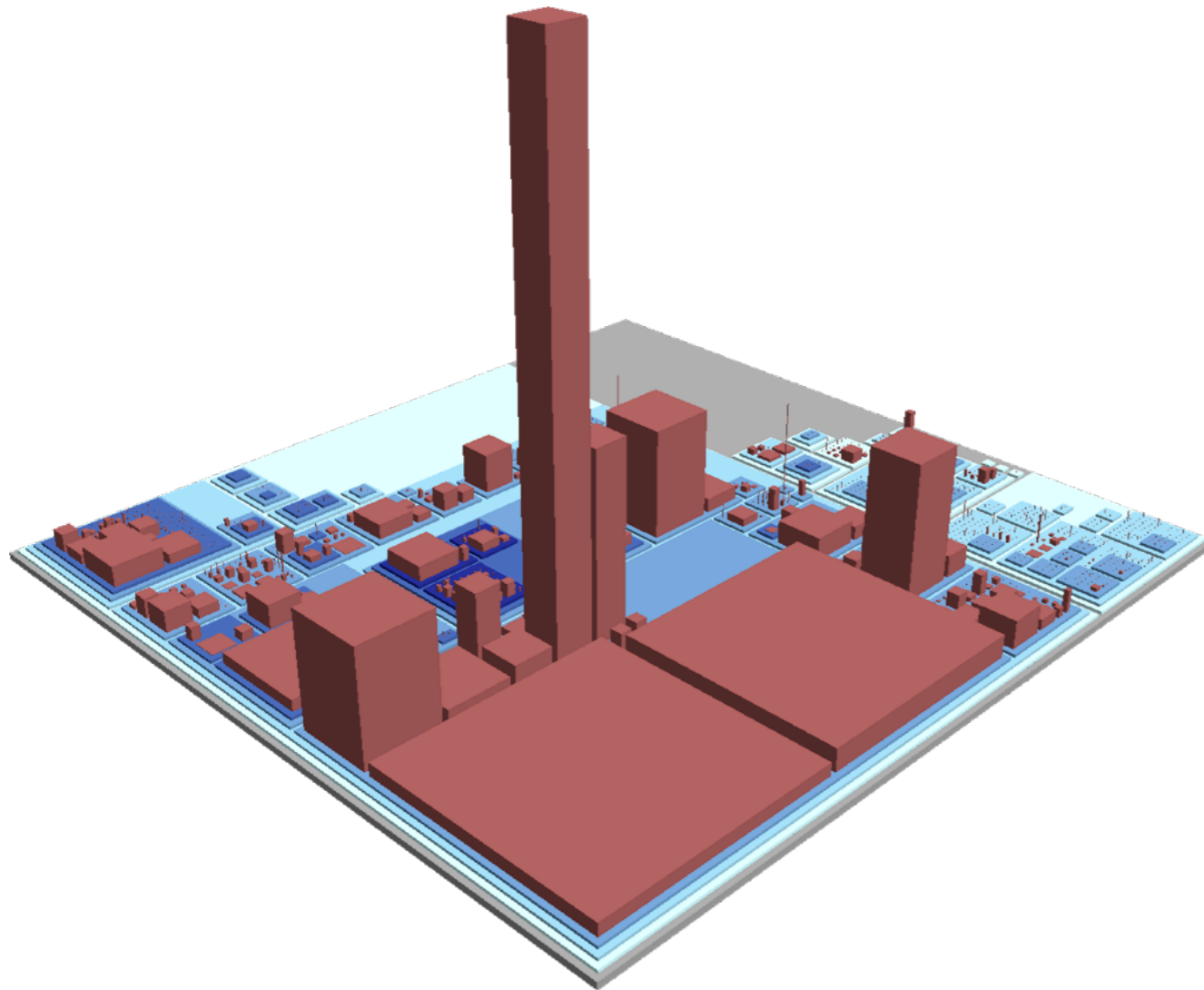
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30

Cityscapes



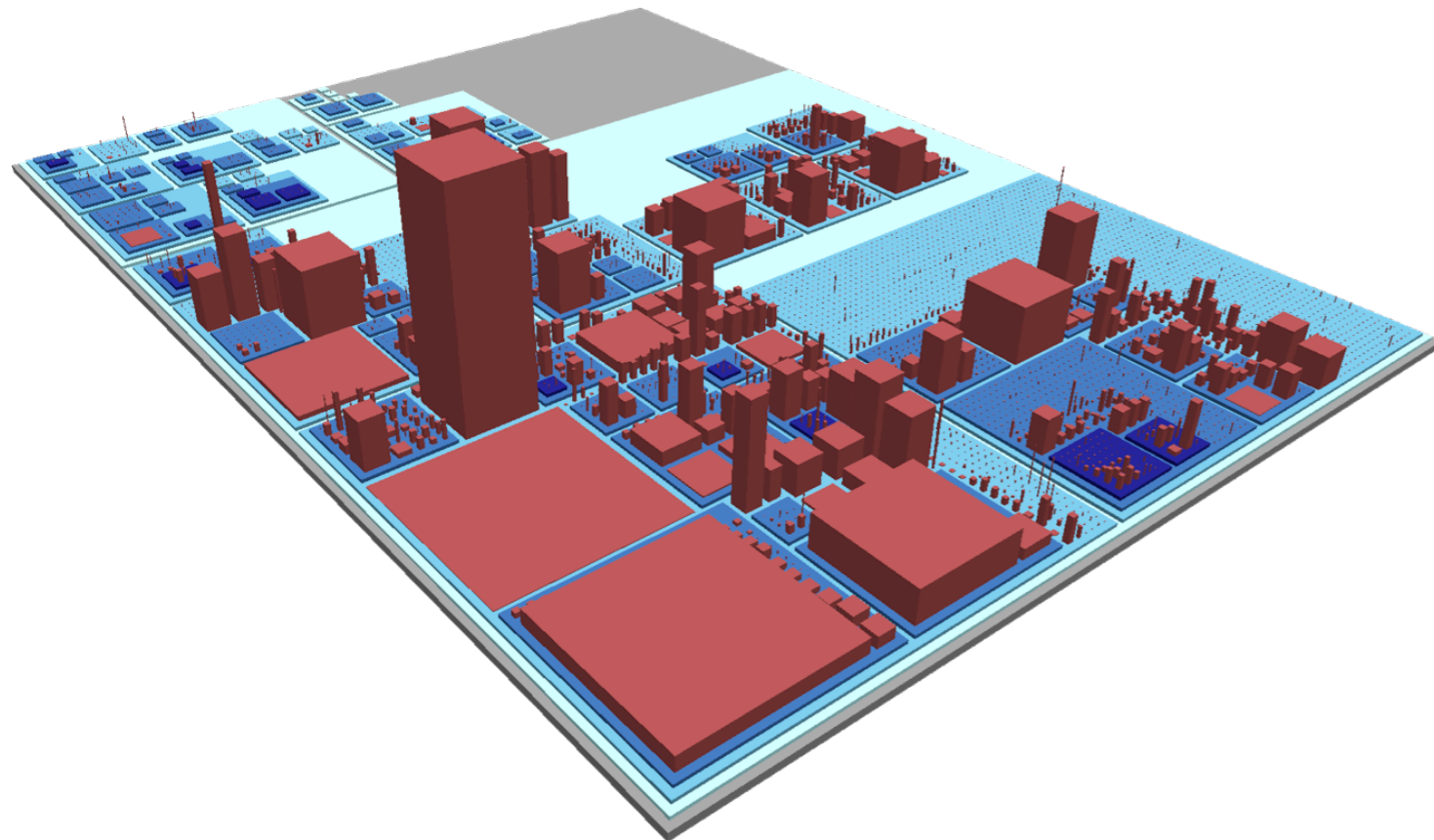
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80

Cityscapes



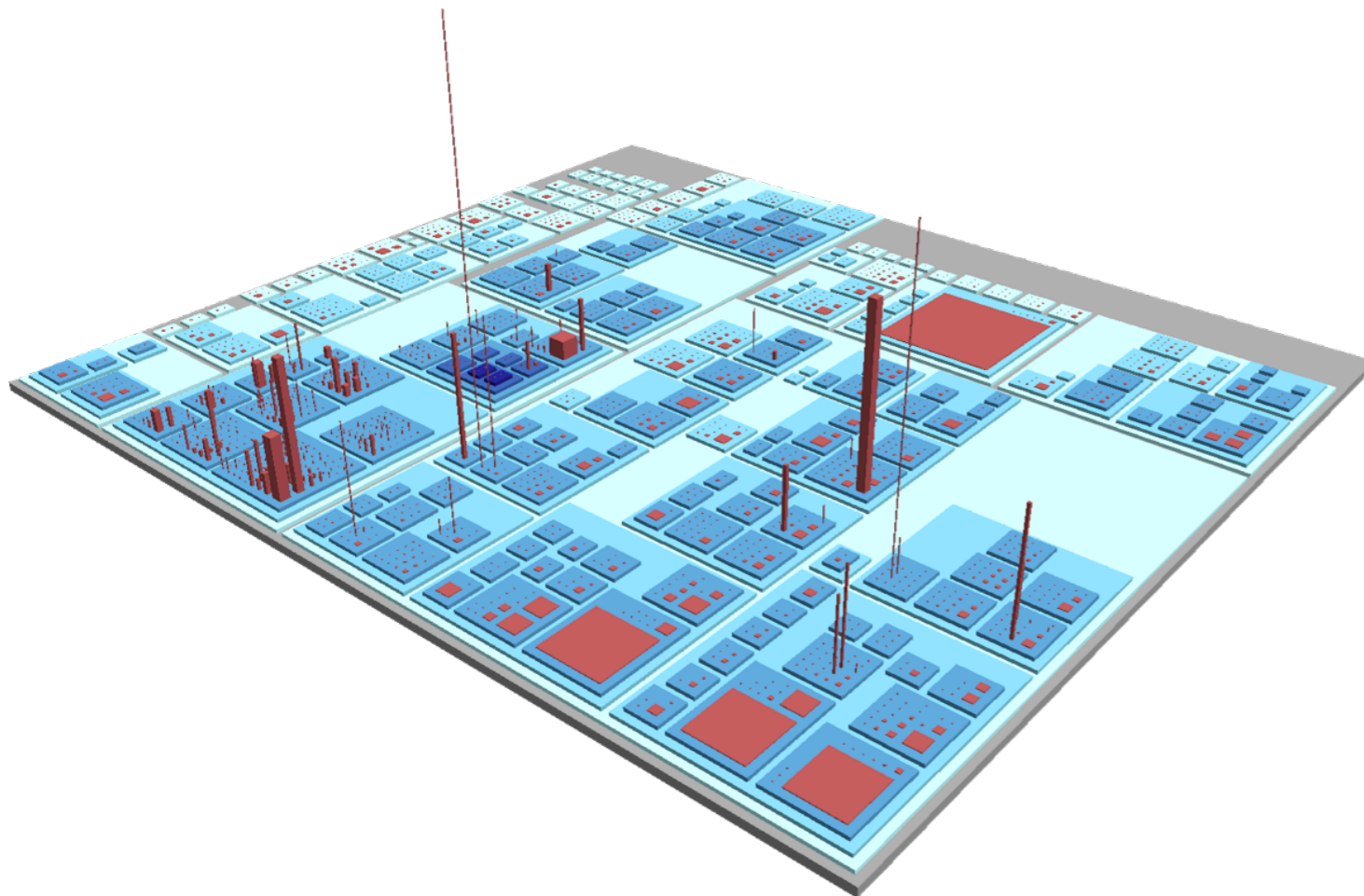
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85

Cityscapes



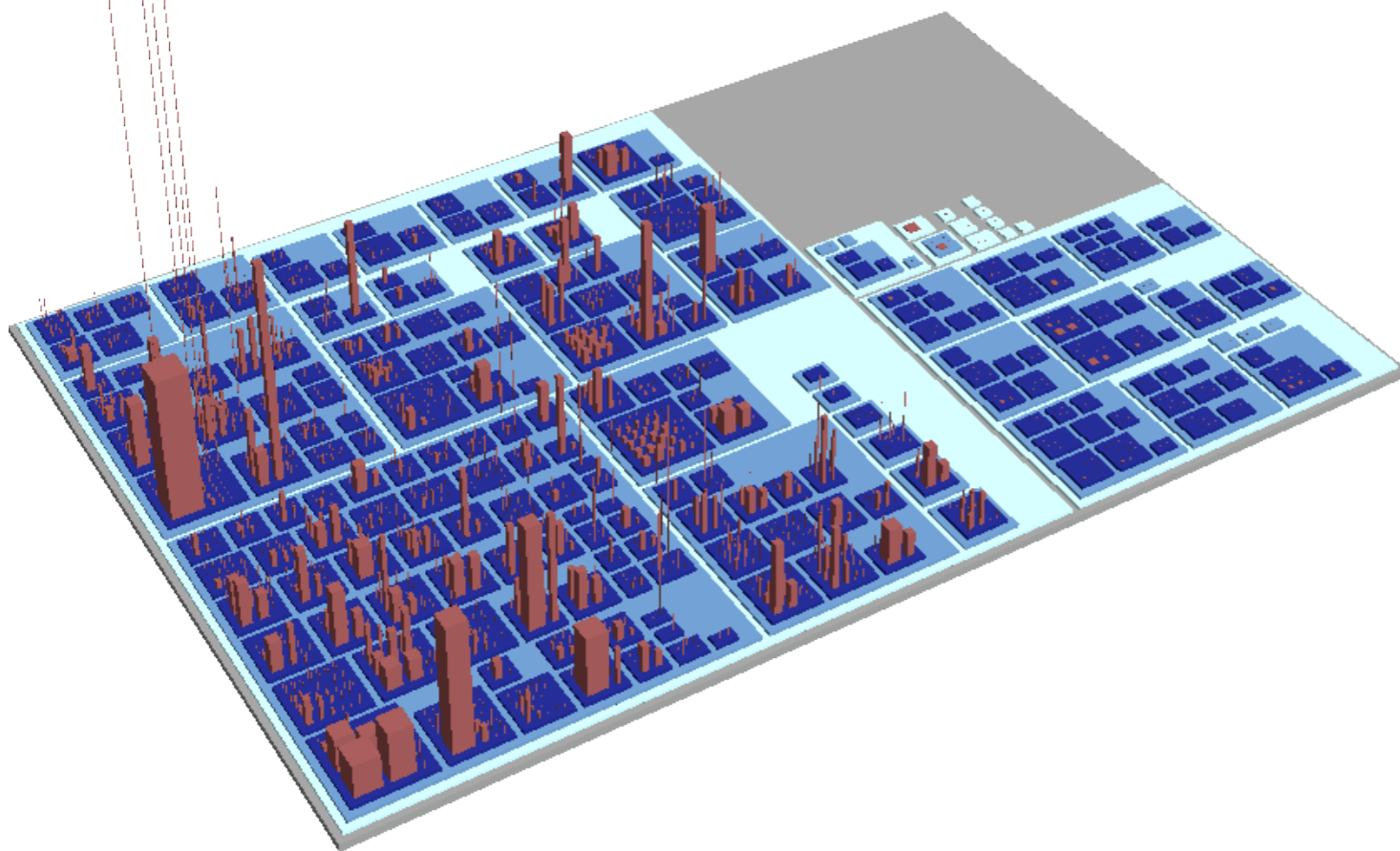
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160

Cityscapes



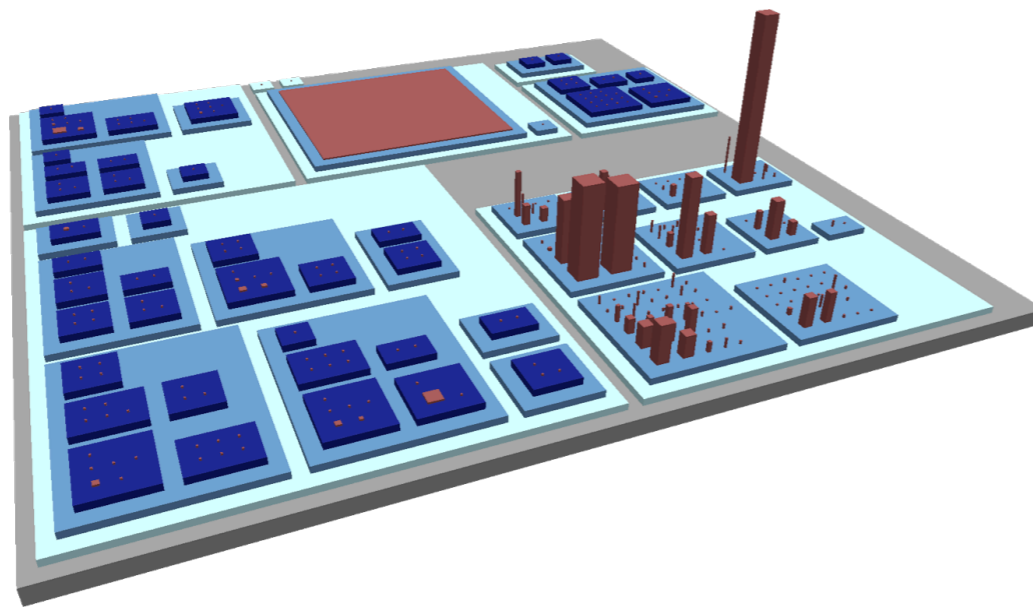
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32

Cityscapes



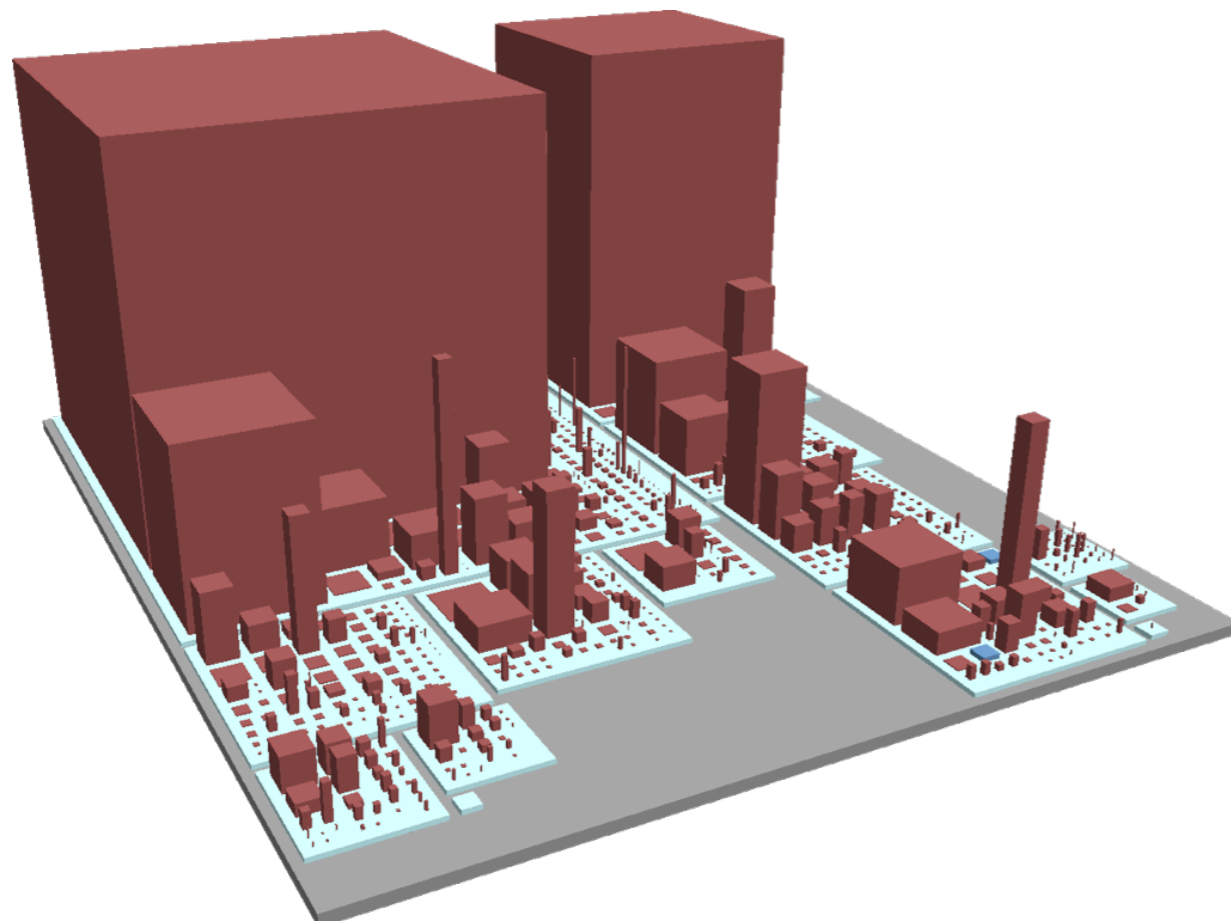
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32
Jun	Smalltalk	288	2236	351

Cityscapes



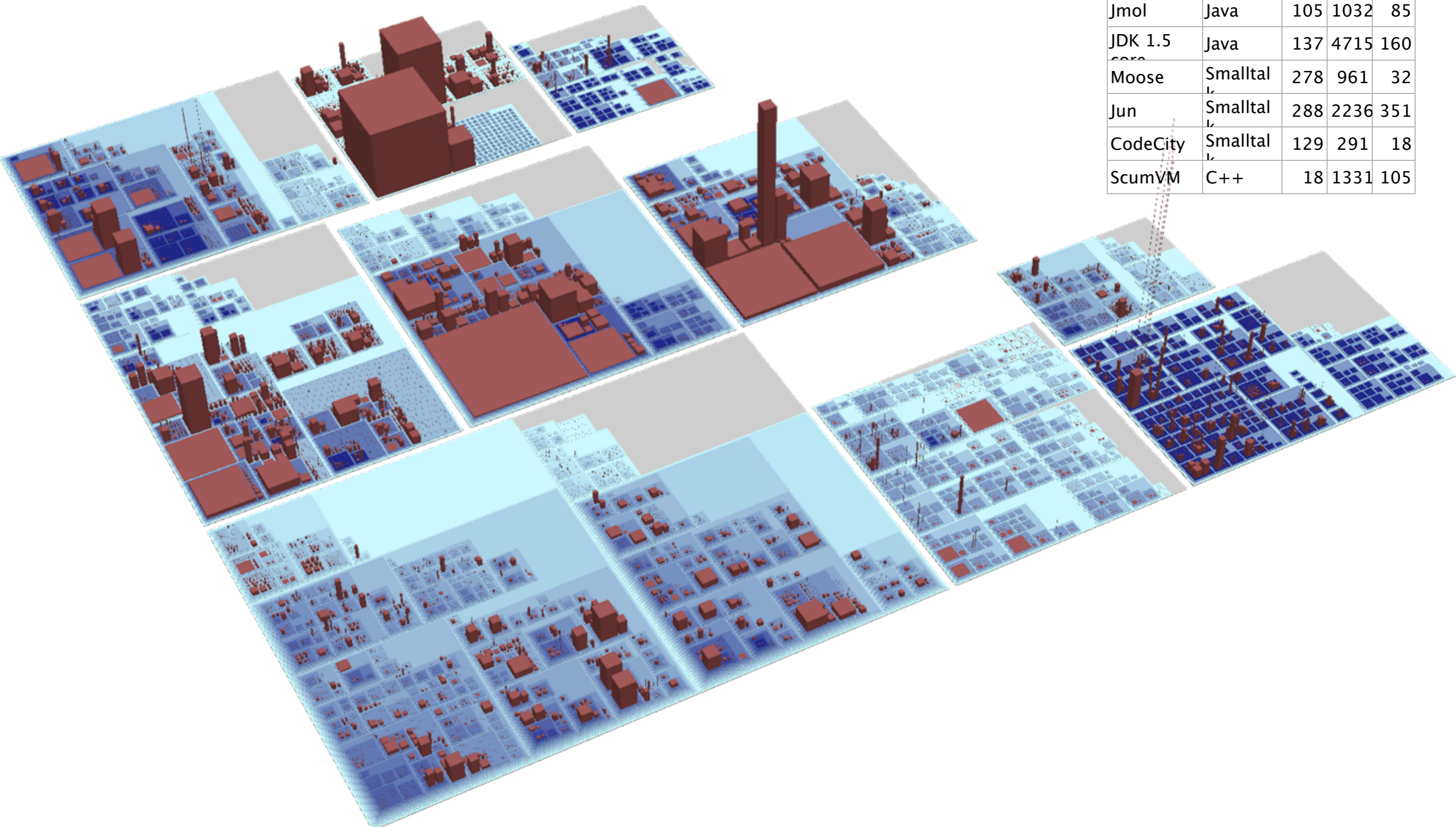
System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32
Jun	Smalltalk	288	2236	351
CodeCity	Smalltalk	129	291	18

Cityscapes



System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32
Jun	Smalltalk	288	2236	351
CodeCity	Smalltalk	129	291	18
ScumVM	C++	18	1331	105

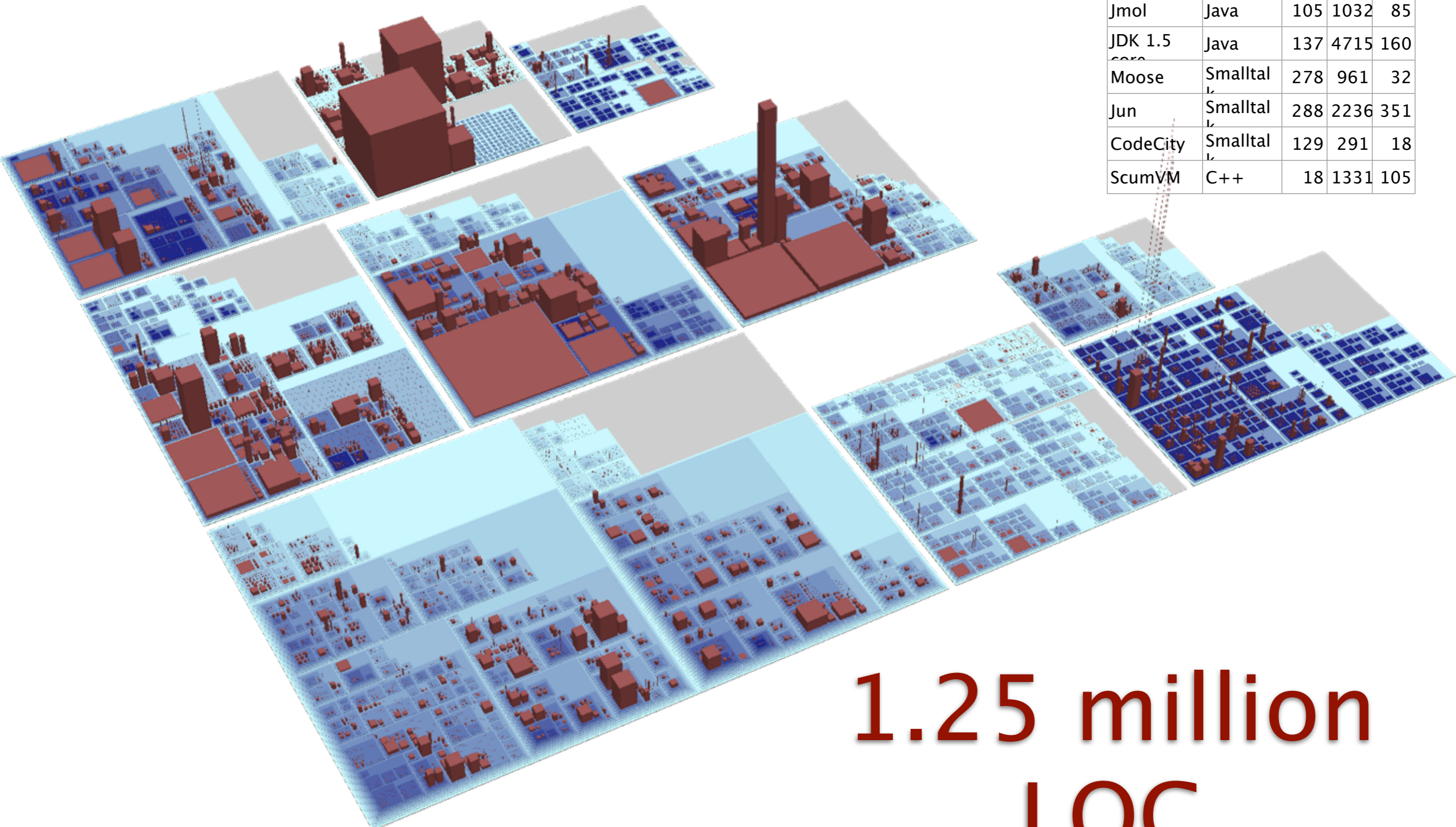
Cityscapes



System	Language	NOP	NOC	kLO
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32
Jun	Smalltalk	288	2236	351
CodeCity	Smalltalk	129	291	18
ScumVM	C++	18	1331	105

Cityscapes

System	Language	NOP	NOC	kLOC
Azureus	Java	457	4734	274
ArgoUML	Java	144	2542	137
JHotDraw	Java	72	998	30
iText	Java	149	1250	80
Jmol	Java	105	1032	85
JDK 1.5 core	Java	137	4715	160
Moose	Smalltalk	278	961	32
Jun	Smalltalk	288	2236	351
CodeCity	Smalltalk	129	291	18
ScumVM	C++	18	1331	105

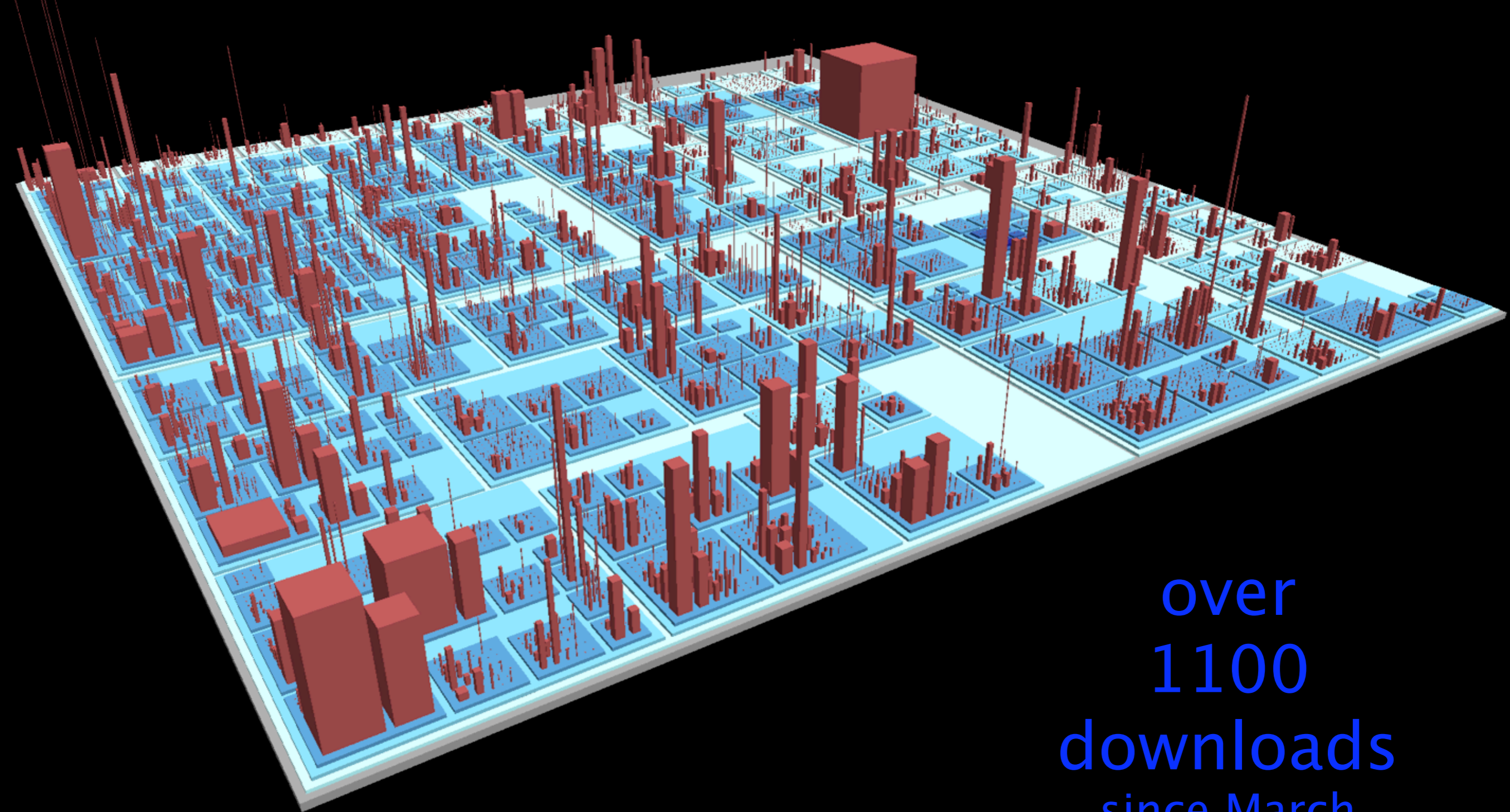


1.25 million
LOC

[http://www.inf.unisi.ch/phd/wettel/
codecity.html](http://www.inf.unisi.ch/phd/wettel/codecity.html)

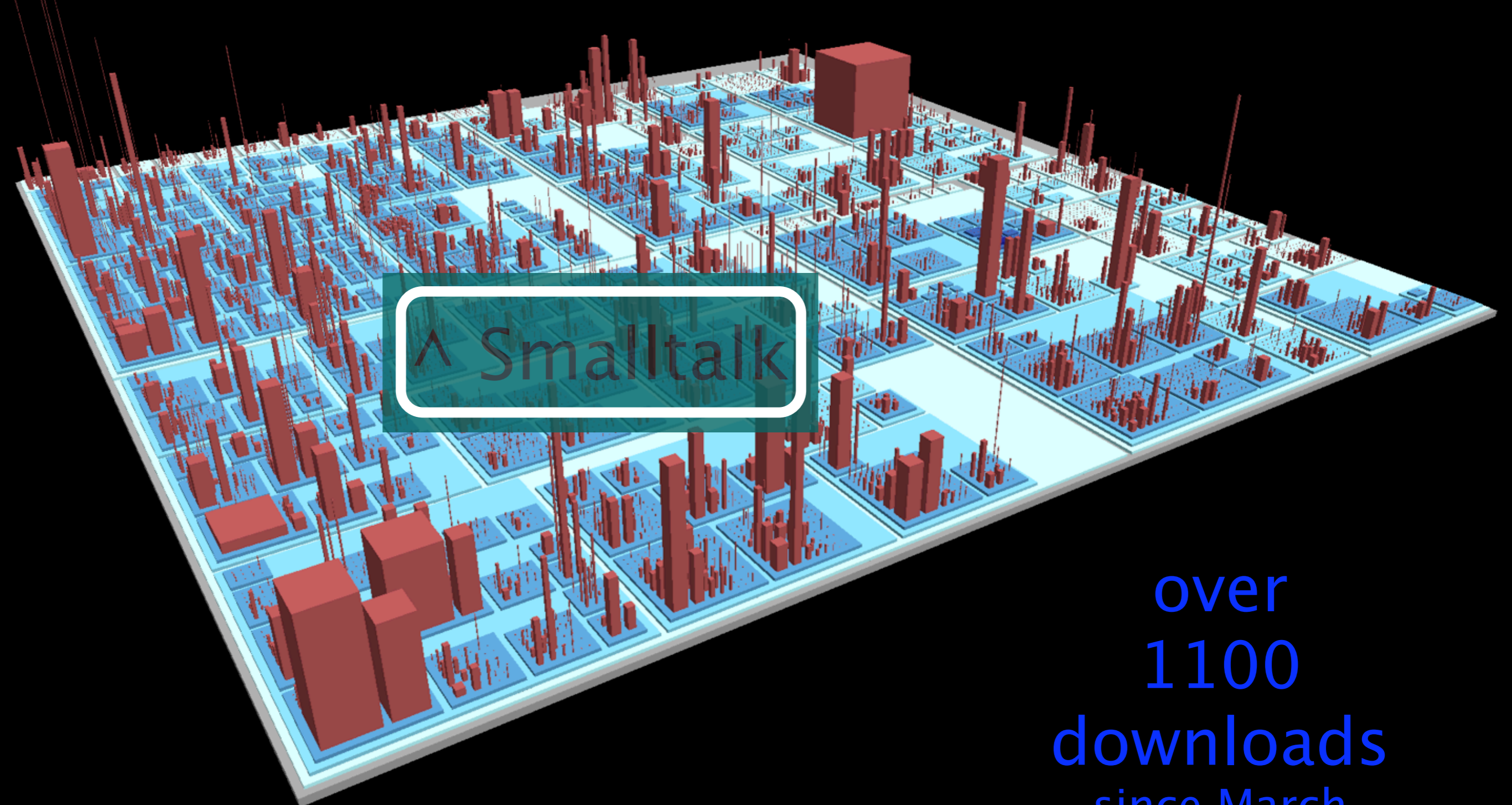
over
1100
downloads
since March
2008

[http://www.inf.unisi.ch/phd/wettel/
codecity.html](http://www.inf.unisi.ch/phd/wettel/codecity.html)



over
1100
downloads
since March
2008

[http://www.inf.unisi.ch/phd/wettel/
codecity.html](http://www.inf.unisi.ch/phd/wettel/codecity.html)



^ Smalltalk

over
1100
downloads
since March
2008

[http://www.inf.unisi.ch/phd/wettel/
codecity.html](http://www.inf.unisi.ch/phd/wettel/codecity.html)